UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

VisuaLeague III: Visual Analytics of Multiple Games

Rafael Nuno Fragoso Afonso

MESTRADO EM INFORMÁTICA

Dissertação orientada por:
Prof.ª Doutora Ana Paula Pereira Afonso
Prof.ª Doutora Maria Beatriz Duarte Pereira do Carmo

2020

Acknowledgments

First of all, I would like to thank professors Dr. Ana Paula Afonso and Dr. Maria Beatriz Carmo as they were always available to support me throughout this project.

Thank you to my parents and sister that were not directly related with the work but gave me motivation, strength and made this possible. Thank you, Sofia Santos for the support, company, and patience during my best and worst moments in this important phase of my life.

To all my friends and colleagues who shared this experience with me, I appreciate the help and moments that made me relax in more stressful times.

Special thanks to Ricardo Costa and all the people who gave their feedback to help improve VisuaLeague's experience in a direct, or indirect way.

To gamers, streamers, game developers and researchers in the area of video games.

Resumo

A informação digital disponível tem vindo a crescer nos últimos anos, sendo fundamental explorar técnicas de visualização que sejam representativas do potencial da informação e que disponham ao utilizador de maneira eficiente os dados que este pertende analisar.

Um caso de interesse são os *eSports* (desportos electrónicos), que se consideram ter uma grande probabilidade de crescimento e para os quais as análises visuais de dados têm um impacto significativo, sendo uma referência para jogadores, treinadores, analistas, espectadores, investidores e programadores. Um dos tipos de jogos mais conhecidos de *eSports* é o *Multiplayer Online Battle Arena (MOBA)* que, tal como muitos desportos tradicionais, têm vários fatores que devem ser considerados num jogo para realizar análise de desempenho. Nos *MOBA*, para além de estatísticas de cada jogo com resultados relativos aos jogadores de cada equipa com totais de dano, pontuações de jogadores mortos, quantidade de dinheiro acumulado, existe também informação relevante das posições dos jogadores (dados espaciais) num período específico no tempo (dados temporais). Podem, também, ser considerados eventos específicos do jogo relativos a diversos objetivos como comprar um item, matar jogadores, destruir torres ou completar objetivos.

Ter uma forma de analisar e visualizar estes dados ajuda tanto programadores e *designers* do jogo, como jogadores, treinadores e analistas. Aos primeiros ajuda a melhorar a jogabilidade ao analisar padrões, tendências e comportamentos esperados ou inesperados. Aos segundos dá apoio à melhoria de desempenho de jogadores avaliando estratégias, comportamentos e respetivos resultados. Este segundo grupo de utilizadores será o público alvo deste trabalho. A informação utilizada será fornecida pela *Application Programming Interface (API)* da *Riot* de partidas de *League of Legends (LoL)*, adicionando ao projeto anteriormente desenvolvido da aplicação *web VisuaLeague II*, novos métodos de análise facilitando interações com múltiplos jogos e pesquisas de equipas. Existe a opção de análise de treinos de jogos profissionais, *scrims*, desenvolvido a pedido de um treinador profissional e informação de equipas profissionais fornecida através de uma *API* de terceiros, *Pandascore*, com o intuito de agilizar pesquisas por jogadores participantes em ligas competitivas de campeonatos variados de *League of Legends*.

Este relatório apresenta trabalho relacionado onde se justifica a relevância dos *eSports* como objeto de análise, bem como dados de telemetria conseguidos através de proces-

sos para extração de informação em jogos virtuais. Foram ainda exploradas técnicas de visualização de dados espacio-temporais utilizadas na área dos video jogos, em particular, os mapas animados, uma das técnicas utilizada no VisuaLeague II. Outros métodos de visualização, nomeadamente processos integrados em jogos virtuais, foram também levados em consideração, uma vez que muitos jogos tomam partido de visualizações dentro do jogo em si, para dar ao utilizador informação relativa a acontecimentos relevantes dependendo do género do mesmo. Por exemplo, representações de *dots* para demonstrar localizações, ou mapas com informação espacio-temporal relativa a jogadores, ou eventos, de modo a dar uma visão geral dos acontecimentos do jogo corrente ao utilizador para prever e planear estratégias. Analisaram-se as suas vantagens e desvantagens ligando-os com a utilização de eventos característicos para cada diferente tipo de análise.

Estas técnicas, quando são considerados grandes volumes de dados, apresentam diversas limitações a nível de desempenho, legibilidade e obstrução de símbolos. De forma a resolver estes problemas, vários processos foram estudados entre os quais se destacam técnicas de agrupamento através de *clustering*. Métodos de densidade como o algoritmo *Density-based spatial clustering of applications with noise (DBSCAN)* são considerados uma boa opção pela flexibilidade em conceber um número arbitrário de grupos e capacidade de percepcionar *outliers*, ou seja, elementos que não pertencem a nenhum dos grupos presentes.

O *League of Legends* é apresentado, indicando-se o que este pode oferecer com foco no tipo de jogo mais relevante a nível competitivo: as disputas de cinco contra cinco jogadores no mapa de *Summoner's Rift*. Explica-se a perspetiva dos jogadores e os objetivos com que os mesmos se têm de preocupar para conseguir bons níveis de desempenho, como por exemplo, tempos para atingir objetivos e destruição de torres. No ambiente competitivo do jogo, há uma distinção entre o *solo queu*, que consiste em quantificar cada jogador atribuindo-lhe um *rank* individual e as equipas competitivas profissionais, que participam em torneios de alta competição. Para perceber qual a informação que é mais relevante extrair, fez-se uma reunião com um treinador profisional. Identificou-se o tipo de apontamentos e o foco da análise, bem como ferramentas de análise utilizadas por profissionais para extrair informação relativa ao desempenho dos jogadores como o *OP.GG*, *Probuilds* e *Mobalytics*.

Posteriormente teve lugar uma segunda entrevista com dois treinadores profissionais de *League of Legends* de modo a identificar as suas necessidades para a análise de jogos e permitir a evolução do protótipo anterior, VisuaLeague II. O protótipo foi desenvolvido numa estrutura *Model-View-Controller (MVC)* com o apoio de diversas tecnologias. O VisuaLeague III suporta três métodos de análise independentes: análise de jogos de um jogador (*Player analysis*), análise de jogos em equipas (*Team analysis*) e análise de partidas de equipas profissionais (*Custom analysis*). Cada uma destas três opções disponibiliza a possibilidade de analisar apenas um jogo ou múltiplos jogos. Os métodos de pesquisa e de

processamento de dados variam em cada caso, sendo necessária a introdução de um nome de jogador (*summoner name*) e região da conta relativa ao servidor para o método *Player analysis*. O método *Team analysis* tem duas vertentes: o utilizador pode criar uma equipa e fazer pesquisas de jogos em que todos os jogadores participaram, ou utilizar informação fornecida pela *API* do *Pandascore*, onde é possível pesquisar por partidas e informação de jogadores competitivos de *LoL*. O método *Custom analysis* permite o acesso a *custom games* com visualizações de jogos para treino de equipas competitivas conhecidos como *scrims*. Na vertente de análise para um jogo, são mostradas visualizações de estatísticas de partidas recentes para obter uma visão geral do desempenho do jogador e é possível utilizar uma visualização focada num mapa animado com dados espacio-temporais, para uma análise mais detalhada.

Na vertente de múltiplos jogos são apresentadas estatísticas e mapas com dados, também, espacio-temporais, agregando a informação de acordo com filtros definidos pelo utilizador. Estes filtros atuam sobre o número de jogos, o tipo de jogo e as personagens (*champions*) escolhidas. As visualizações obtidas podem, também, ser customizadas através da interação do utilizador escolhendo, dos parâmetros disponíveis, que dados devem, ou não, ser mostrados. É possível focar a análise num determinado tempo de jogo e agrupar as posições representadas através de um algoritmo de *clustering*, *DBSCAN*, de modo a tornar a visualização legível quando muitos dados estão a ser visualizados ao mesmo tempo.

Realizaram-se testes de avaliação com dois treinadores profissionais, onde os participantes usaram o VisuaLeague III e responderam a um questionário. A ferramenta recebeu críticas maioritariamente positivas na análise de *custom games* e dados espacio-temporais para múltiplos jogos. O primeiro por ser o único processo conhecido para analisar este tipo de partidas e o segundo pela capacidade de trazer uma análise de jogadores capaz de tirar conclusões tanto na fase de jogo como na de preparação de equipas. O método de pesquisa por equipas foi considerado o menos útil, pelo facto dos jogadores normalmente não praticarem juntos. Foi sugerida uma pesquisa por equipas onde, em vez de considerar um grupo de pessoas, deveria ser realizada a pesquisa por contas da *Riot*, assim seria possível pesquisar por diversas contas de um, ou vários, jogadores.

Finalmente, mencionaram-se as possibilidades de trabalho futuro sugerindo vários aspetos onde o VisuaLeague III poderia melhorar. Por exemplo, alterando a forma de realizar a pesquisa por equipas, incluir filtros e visualizações direcionadas a items, pesquisas de agregações por tempo e opções de filtros por papel de jogador, principalmente quando não é possível identificá-lo nos dados, como é o caso dos *custom games*.

Palavras-chave: Visualização de dados de jogos, dados espacio-temporais, visualização de dados agregados, visualização analítica, *League of Legends*

Abstract

Digital data available has been growing over the last years and with it, the need to create representative ways to understand and make use of its potential with visualization techniques that can be applied in different purposes.

One of these cases are eSports (electronic sports), considered nowadays a sport with high growth expectation, and for which data analyses can have a significant impact. One of the most popular game type practiced in eSports is the Multiplayer Online Battle Arena (MOBA) genre represented by one of the most popular competitive games, League of Legends (LoL), which will be the case study for this thesis. As many traditional sports, there are various events to have in consideration when observing performance of gameplay. In addition to statistics for each game there is relevant information on players' positions (spatial data), in a specific period in time (temporal data). Specific events in a game, related with objectives, can also be considered, such as purchasing an item, player kills, destroying towers, or complete objectives.

Having a way to analyze and visualize this data helps not only programmers and game designers to improve gameplay but also players, coaches and analysts to improve player performance.

The objective of this work is to redesign the previous prototype VisuaLeague II, and propose a new version, VisuaLeague III in order to explore techniques to implement analysis for multiple games, team searches and access to professional games' training sections, scrims. Common problems presented in the analysis with voluminous amount of data, like cluttering and overlapping, are addressed by adding filters to searches, interaction with the visualizations, aggregation of data, and clustering.

The developed prototype, VisuaLeague III was evaluated by professional coaches to understand if the searches and visualization techniques implemented are adequate for analysing players' performance in a competitive environment. The results demonstrate overall positive attitude with particular interest in analysis for custom games and multiple games analysis as those provide visualizations that do not exist in common tools, specially, regarding spatiotemporal data.

Keywords: Game data visualization, spatiotemporal data, aggregated data visualization, visual analytics, League of Legends

Contents

List of Figures

Acronyms

AOIs Areas of Interest. xv, 11

API Application Programming Interface. v, vii, xvi, 2–4, 6, 20, 24, 26, 28–30, 32–35, 37–39, 41–45, 47, 51, 71, 74–76

APOC Awesome Procedures On Cypher. 35

CS Creep Score. xvii, 55, 57

CSS Cascading Style Sheets. 36

DBSCAN Density-based spatial clustering of applications with noise. vi, vii, 12, 13, 15, 34, 36, 60, 61, 73

Eps Epsilon value for DBSCAN. 13, 36, 60, 61, 70

FPS First-Person Shooter. 1, 14

HTTP Hypertext Transfer Protocol. 26, 34, 35

JS Java Script. 34, 36

JSON JavaScript Object Notation. xvi, 6, 7, 34, 35, 37, 39, 40, 42, 43, 47

KDA Kills, Deaths and Assists. xvi, 20, 28, 29, 48, 49, 54–57, 64, 66, 67

LCS Longest Common Subsequence. 11

LoL League of Legends. v, vii, ix, xv, 8–10, 15–17, 19, 25, 28–30, 39, 45, 47, 52, 57, 59, 63, 67, 70, 71, 73, 74

LP League Points. 19, 49

minPts Minimum points to be considered a cluster. 13

MMR Match Making Rating. 19

MOBA Multiplayer Online Battle Arena. v, ix, 1, 17, 20

MVC Model-View-Controller architecture. vi, 24, 32, 34, 35

OPTICS Ordering points to identify the clustering structure. 13

OS Operating System. 34

RPG Role-playing Game. 14, 17

RTS Real-Time Strategy. 1, 5–8, 14, 17

ST Spatiotemporal. 7–9, 15, 26, 28, 59, 68, 71, 74, 75

URL Uniform Resource Locator. 39, 43, 45, 52, 54

Chapter 1

Introduction

This chapter introduces the main motivation of the thesis addressing why video game analysis and League of Legends are relevant as a research subject. The goals and contributions of the project are characterized, followed by the structure of this document with an overview of its content.

1.1 Motivation

Games were always a common hobby to people. Nowadays virtual games are a tendency for the young generation who start to immerse this world very early in life, playing or spectating streaming websites like Twitch (Hamari and Sjöblom, 2017). The concept of eSports (electronic sports) is growing every year (Chikish, Carreras, and Garci, 2019) as a worldwide phenomenon for professional players to compete in tournaments using electronic devices namely computers, consoles and phones (Olsen, 2015). Gaining the attention of many, this group of enthusiastic events has got its place not only in the internet but even in U.S. television broadcasts (Holden, Kaburakis, and Rodenberg, 2017). Many types of games can be found in eSports being the most well known focusing on Multiplayer Online Battle Arena (MOBA) like League of Legends, First-Person Shooter (FPS) like Counter-Strike, and Real-Time Strategy (RTS) like Starcraft II (topEsports, 2019). The constant growth of players and spectators contributes to eSports' popularity bringing a need to keep improving gaming and spectators' experiences.

To people that only like to watch eSports, following the media to watch key moments and socialize is a big enough motivation. However if we talk about professionals and developers who want to improve their game, a way to better understand what's going on with the game and its audience is essential. To achieve, this multiple methods can be considered like questionnaires to the community, play-testing or videotaping (Schertler, Kriglstein, and Günter Wallner, 2019)(Günter Wallner, Halabi, and Mirza-Babaei, 2019), but those are subjective, time consuming and sometimes incomplete (Moucho, 2018). Other method available nowadays is relying on telemetry data which is automatically created while playing the game by saving the interactions of the player. It gives direct information of concrete events that happened and can be further analysed with the amount of detail needed.

Telemetry data gathered can, however, be very extensive, leading to voluminous spatiotemporal datasets of information and becoming hard to analyse (He et al., 2019). The main goal of this thesis is to address this problems by exposing visualization techniques that better materialize users' needs to analyse telemetry data gathered from Riot Application Programming Interface (API) to improve player performance. The study will focus on League of Legends' professional environment where coaches and analysts train a team or group of players by analysing behaviour, tendencies and results, but it could also be relevant for casual players or developers to realize strengths and weaknesses of their game's balance. Studying this kind of data can give detailed information about players' actions at a specific time and place or general information of the team choices. To help any of those cases it is essential to find visualization techniques for spatiotemporal data analysis in a practical and efficient way capable of adapting to each user's needs for one or multiple matches.

There is already a prototype of the work done by Moucho, 2018 and Vieira, 2017 who created a web application to show League of Legends' telemetry data available through Riot's API, VisuaLeague II, focusing on casual players interests in searches for one player (summoner). VisuaLeague II uses the animated map to show players' movement, and static data visualization techniques, such as charts with statistics, enabling player analysis and teams comparison.

The purpose of VisuaLeague III is to find ways to overcome previous limitations, namely add a capacity to analyse multiple games, enable team searches and grant access to custom games, while improving the ability to make comparisons and spot tendencies of players or teams. Common problems presented in the analysis with voluptuous data, like cluttering and overlapping, are addressed by adding filters to searches, interaction with the visualizations, aggregation of data, and clustering.

1.2 Objectives

Using telemetry data collected through Riot's API and the prototype developed by Moucho, 2018, the main objective of this work was to extend the VisuaLeague II, in order to give a wider variety of choices to professional coaches and analysts visualise data about players, or teams, while being useful for personal improvement in both, competitive, and casual environments of the game. Additionally, it is necessary to reimplement methods that already became obsolete in consequence to recent updates, and new functionalities of the Riot API. Although there are other games that also provide this type of data, League of Legends was chosen as case study due to its popularity, accessibility and tendency to grow as a game. Its constant improvement also keeps relevant new analysis of data, since every season new updates are introduced in game with big impact on gameplay and trending strategies, giving always something new to analyse and improve.

The main objectives of the thesis can be enumerated as follows:

- Understand previous work and adapt VisuaLeague II to make its functionalities full operational with the recent updates of the Riot API;
- Extend the current visualizations with the addition of charts, user interaction, customization options, and new searches for teams and professional matches;
- Propose interactive visualizations to analyse multiple games or multiple spatiotemporal variables;
- Test and evaluate the prototype with expert users.

1.3 Contributions

The main contributions of this work can be enumerated as follows:

- Study and discussion of previous research focused on relevance of eSports, virtual games and resultant telemetry data gathered granting the ability to create visualizations for different analysis types for different games' genres and respective objectives emphasizing on player performance and spatiotemproal data.
- A deeper understanding on League of Legends' gameplay in a competitive environment through professional coaches interviews with insights about their analysis process for player performance in Summoner's Rift matches for ranked solo queue and tournaments.
- Implementation of VisuaLeague III, redesigning the a previous prototype and adding three new analysis methods for single player, teams and custom games, with options for single and multiple match analysis available for each one. Resultant visualizations change according to user filters and analysis type, making use of different techniques like variant chart types for statistical information, and maps with timelines and clustering for spatiotemporal data.
- A user study with two professional coaches addressing effectiveness of the resultant prototype as a tool for scouting and analyse player performance according to League of Legends' competitive environment needs for each of the available analysis types in single and multiple matches.
- Writing of an article developed together with Ana Paula Afonso and Maria Beatriz Carmo describing the developed prototype.

1.4 Structure of the document

Chapter 2 shows related work about eSports' relevance nowadays, introducing the concept and enforcing the motive to choose virtual games as an analysis focus. After that, telemetry data is defined and visualization techniques applied to video games are discussed. Those include spatiotemporal data visualizations and methods of displaying aggregated and individual data including advantages and disadvantages of each one. Diving more into games, some examples of previous visualisations already tested are shown giving examples for different game genres.

In Chapter 3 a more deep look into League of Legends' specifications shows how the game works by summarizing its objectives and giving an insight on the competitive environment. Current solutions used to analyse its telemetry data are also referred as they're the tools coaches, players and analysts rely for their analysis. The previous prototype VisuaLeague II, is described together with the technologies used to accomplish the intended objectives and ways to process data for its main visualization, the animated map.

Next in Chapter 4, VisuaLeague III is fully described starting by showing an interview with two professional coaches that determined new requirements for the application. This chapter starts with a small overview of the prototype to then introduce the new architecture with the different models, views, databases and APIs used. Next new ways of processing data for single player, teams and custom match searches for one and multiple matches analysis are presented, being then complemented by the application's design together with the resultant visualizations for each one.

In Chapter 5 user tests done with two professional coaches are provided and discussed for each analysis type individually.

Finally, Chapter 6, shows conclusions and possible improvements for a future work according to the new acknowledgements obtained through this project and coaches' feedback.

Chapter 2

Related Work

This chapter presents research on several topics related with the project. Starting with Section 2.1 where a brief introduction to eSports' state of art helps to understand the concept and why it is relevant to analyse games data, together with the growing potential for the near future. Then telemetry data is described in Section 2.2, showing how it relates to games and its potential. In Section 2.3 is presented an explanation of the data visualization concept and posteriorly complemented for spatiotemporal data in Section 2.3.1. Section 2.3.2 shows the importance of aggregated visualisation techniques in order to understand advantages and disadvantages of each approach. Also, Section 2.3.3 presents visualization techniques used in games.

2.1 eSports

There is no consensus nowadays whether eSports should be considered an actual sport, but it is likely to be officially accepted as one in the near future, as stated by Stefani, 2017 and Hallmann and Giel, 2018. Since it does not measure player's competence by their physical prowess, it is better to be seen as a sport of the mind, since it focuses on player's capacity to analyse situations and make the best choices enforcing a necessity for reactive skills to be faster than other players. Thompson et al., 2013 defends that some genres of games can be related with specific sports like, for instance, the RTS genre is often compared with chess, due to its high requirement in tactics and strategy. It is undeniable the impact that eSports currently has in entertainment being a successful professional career to many as well as showing a pro-eminent future with the highest growth expectation, replacing football (Chikish, Carreras, and Garci, 2019), (Journal, 2019). Figure 1 shows eSports worldwide market revenues and audience. Although not very high in 2014, in 2021 it is expected to have more than twice the spectators and more than eight times the market revenues.

A way to analyze real time strategies is to watch match key points of professional matches streamed from eSports. Understanding players behaviours and choices can be

Figure 1: eSports worldwide market revenues and audiences (Chikish, Carreras, and Garci, 2019)

useful to improve the performance of an individual.

Analysts and coaches also use streamed information to train their teams and developers can understand their next step of improvement to the game's gameplay.

2.2 Video games telemetry data

Telemetry data is captured automatically using real-time instruments and sensors. Its format and content may differ depending on the analysis focus. In the case of video games is relatively simple to get this data by saving information about what is happening to a database where it can be accessed and analysed afterwards (Afonso et al., 2019). As it is possible to define player tendencies and behaviours, not only players benefit from this analysis but also developers of the game can have a better insights of their implementations' impact to make improvements. Although the focus of these work is about virtual games it is important to mention the usefulness of telemetry data outside of this spectrum, for example in satellites (Skobtsov et al., 2017), analysing species habitats (Rogers and White, 2007), (Hu et al., 2020), or simply to guide a person through a map, like Google maps. Thompson et al., 2013 makes a comparison between the skills required to play RTS games and surgery, defending that telemetry data in video games is also a good source to test expertise variables in other affairs.

Nowadays, many games store and provide telemetry data for different purposes, sharing it through APIs which accept requests on data from an user and responds in a specific format, namely JavaScript Object Notation (JSON). Two very similar games in this regard are Dota2 and League of Legends of the same genre. Both share a very close way of handling this sets of data, but also more distinct games like Counter-Strike: Global Offensive and Fortnite fall into this same approach. Examples of the JSON file data are presented in Figure 22 and Figure 23 in Section 4.6, related with data processing for VisuaLeague.

A typical way to access this data is search for a player and retrieve events related to the game's genre. There are different ways to utilise this kind of information and a very common approach is from a web application, such as the approach followed in VisuaLeague (Section 4.5.1). In Section 3.3 some details about their information are exposed in order to demonstrate their importance for current player and coach's analysis.

2.3 Data Visualization

People use sight as a main source to gather information. Information is everywhere, used as a medium for several contexts and consumed by humans in their everyday life, be it outdoors or while surfing the internet. This involvement makes visualisation techniques a reliable approach when representing digital information (Copeland et al., 2018). Created from different sources, constituted of diverse types and stored in many formats, the amount of data that we have at our disposal nowadays is voluminous and still growing, which makes finding the best way to extract knowledge from these information not trivial as it heavily depends on the intentions of the one who's interpreting it (Günter Wallner, 2019).

Section 2.3.1 describes spatiotemporal data and presents common visualization techniques in video games. Section 2.3.2 shows the importance of providing aggregated data for analysis and different approaches to achieve those results. Section 2.3.3 describes examples of visualizations used inside the game.

2.3.1 Spatiotemporal data visualization

Spatiotemporal (ST) data is a representation of data with space and time attributes. The first, space, can be considered as geographical coordinates of latitude and longitude values. The second, time, is usually referred by hour, minute and second, or determined by an interval. ST data is usually found in events, geo-referenced variables, geo-referenced time series, moving points and trajectories (Shi and Pun-Cheng, 2019).

Visualizing space information has always been dominated by maps. Maps are a theme based graphic representation of spatial concepts (*History of Cartography* 2017) usually used in video games. Analyzing this type of data can be useful for video game analysis to improve player decisions and gameplay, or even during the programming process to improve game design (Schertler, Kriglstein, and Günter Wallner, 2019).

Kuan, Wang, and Chuang, 2017 shows a system to visualize important battles in the popular RTS genre game, StarCraft II. They analyse movement of the armies being represented as stains of ink related with player's team colour and battles are curved arrows simulating movement. Figure 2 shows a representation of two different views in their visualization system. Figure 2 (a) is a color-coded chart that represents nations economics. It is an RTS genre with multiple unit types therefore Figure 2 (b) indicates time that technologies and buildings are constructed. Figure 2 (c) is a death theme river that shows deaths over time. Figure 2 (e), (f) and (h) are maps representing different elements of the game while giving spatial context.

Figure 2: System to visualise StarCraft II matches (Kuan, Wang, and Chuang, 2017).

Animated maps are a popular form of visualization in games due to their ability to present content over time what gives an adequate method to convert ST data to users in the context of player performance for analysis in video games (Vieira, 2017). Studies regarding the importance of animated maps can be found in Gonçalves et al., 2018 and Moucho, 2018. An example of one animated map serving as a support tool to give a spatial notion while the main focus is on timeline events can be found in Figure 3 e).

The data recorded in video games usually includes ST data, like players' movement and other ST events in which they participate like kills, in multiplayer games. Agarwal, Günter Wallner, and Beck, 2020 created a tool displaying data over time, where the main filters act upon this events, letting the user identify the most relevant and remove the others if they are not needed. This filters are shown in Figure 3 b), being useful to remove cluttering that can emerge from large data sets representations without the need of aggregation techniques.

Another example of visualizations of events can be found in Mejorado and Marquez, 2019 work where League of Legends (LoL) is also used as a case of study. The authors propose two visualizations to explore player interactions (Figure 4 left) and accumulation of objectives (Figure 4 right) in order to identify team behaviour and collaboration. In the visualization of player interactions, colors identify the team, where blue is the winning team and red the losing team. The circles refer to each player and size represents the amount of gold collected. The arrows represent an interaction between players of the same team (assists), or enemy team (attacks). For the accumulation of objectives visualization, the center node color represents a team and other colors are used in the remaining circles for each objective. Objectives are illustrated in the circumference position, starting from

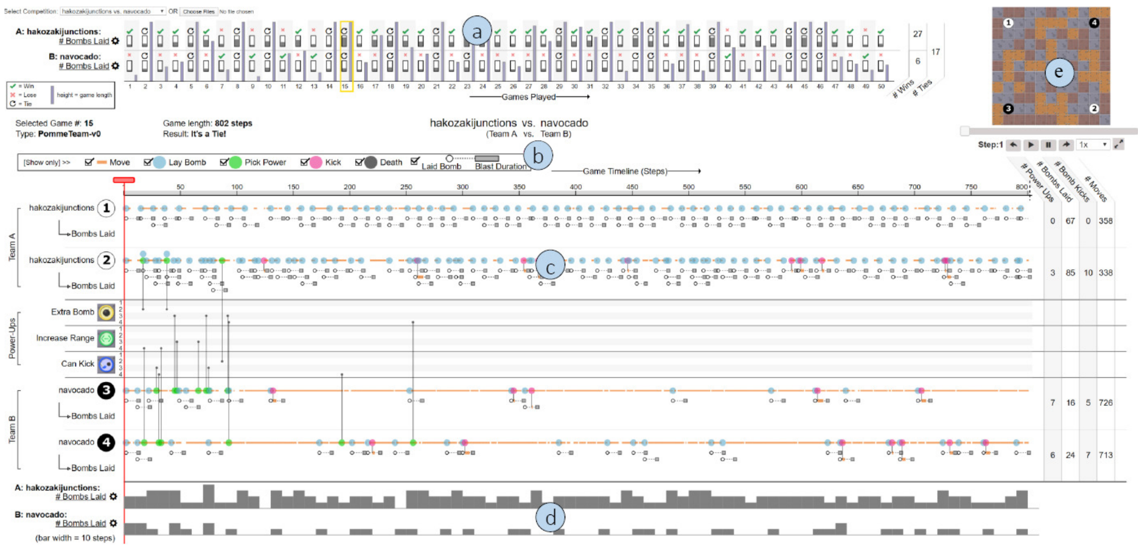


Figure 3: PomVis interface from Bombalytics (Agarwal, Günter Wallner, and Beck, 2020)

the angle 0º in the middle right, representing minute 0, and finishing in the angle 360º, representing the match length.

Although interesting, this work does not present ST data related to matches and does not give the ability to analyse data from recent matches done by specific players or teams. The player interaction visualization gives an idea of how players interacted, but it does not give context of the match, making it difficult analyse player performance. The blue and red colors represent the winning and defeat teams, but the same colors can be related to the map side (also blue and red) which can cause confusion in analysis. In the accumulation of objectives visualization is possible to identify which of the team was faster to complete objectives and their respective order. However, it is not possible to know if the objectives were completed in the right time since the exact time is not presented. The fact that every match is represented from angle 0º to 360º can cause misinterpretation in time, since different matches have different lengths.

Section 3.1 describes LoL to better understand this concepts.

2.3.2 Aggregated data visualization

Large dimensions of trajectory datasets can cause serious problems for most visualization techniques, namely, presenting cluttering and overlapping of the symbols, obstructing information perception and analysis of patterns (visual effectiveness problem) (Guo et al., 2006), (Gonçalves, Afonso, and Martins, 2015), (Günter Wallner, Halabi, and Mirza-Babaei, 2019). Those problems are a major concern when analysing games, as usually it is important to consider multiple information for multiple players (Günter Wallner, Halabi, and Mirza-Babaei, 2019), (Schertler, Kriglstein, and Günter Wallner, 2019), (Günter Wallner, 2019). Common approaches to solve these problems are filtering (Tobler, 1987),

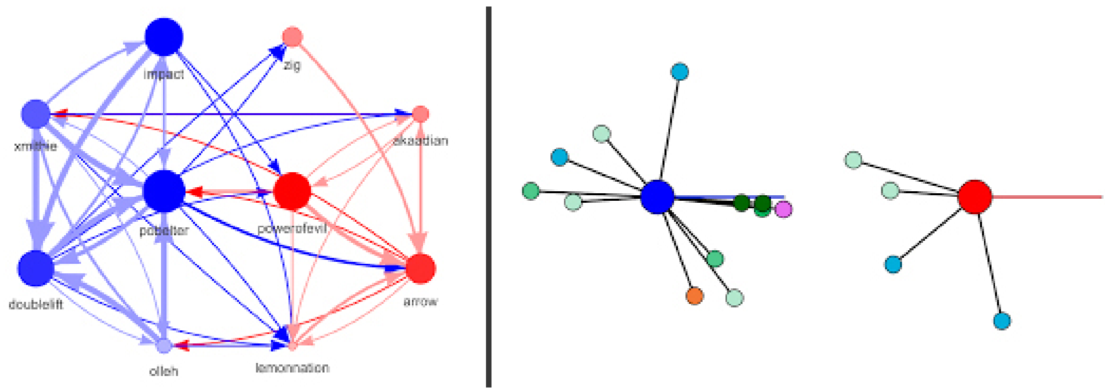


Figure 4: Example of visualizations to team performance analysis in LoL. Left: Interaction visualization. Right: Accumulation of objectives visualization (Mejorado and Marquez, 2019).

clustering and data aggregation (G. Andrienko and N. Andrienko, 2008).

Some authors proposed solutions where it is possible to have individual data visualizations and aggregated data visualizations (Figure 5) to understand the advantages of each approach. (Günter Wallner, Halabi, and Mirza-Babaei, 2019). This work represents player trajectories in an Infinite Mario map (Markus Persson, 2008) considering arousal levels and events. The two figures at the top show data with individual player trajectories making lines for each trajectory with colors representing the arousal and icons representing events. In the two bottom figures, data is displayed aggregated, instead of representing each event. In this case, the size of an icon gives the idea of the quantity, being bigger related with more events. The thickness of the lines is used to represent the quantity of movement in that trajectory, and thicker lines means a lot of players passed in that place or a player passed there a lot of times. The map was divided in areas creating a Voronoi tessellation and the saturation of colour gives a notion of the arousement level in each area resembling a choropleth map.

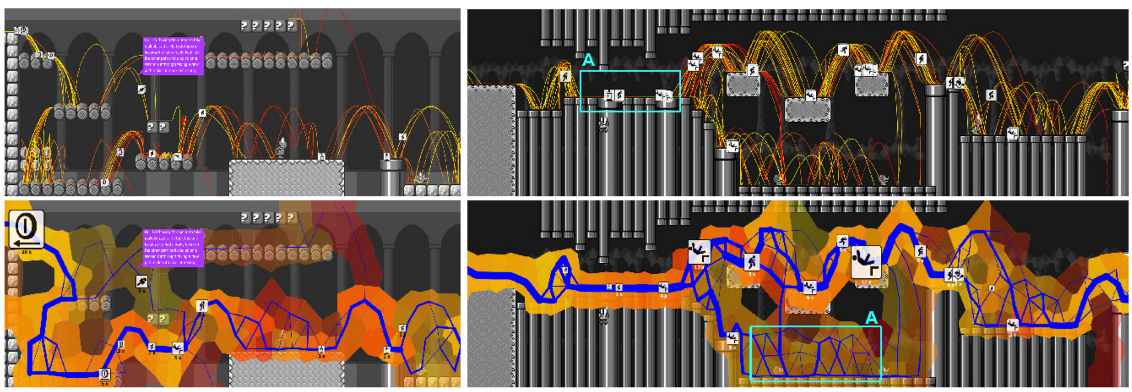


Figure 5: Infinite mario player behaviour visualization (Günter Wallner, Halabi, and Mirza-Babaei, 2019). Top: Non aggregated visualization. Bottom: Aggregated visualization

By looking at both images is possible to tell that the aggregated visualization gives a more instant readability as it is easier to see which places had more or less of each category levels. on the other hand it is impossible to know exactly how many. The best approach most likely depends on the purpose of the analysis and also on the quantity of data gathered. To overcome this problem, in a more recent work, Agarwal, Günter Wallner, and Beck, 2020 implement both options in a harmonious interface, with a view showing aggregated information (Figure 3 a)) and views with individual data for each game (Figure 3 c) and d)).

We are going to describe some common approaches to visualize the results for aggregation of movement data applied to game data.

Semantic trajectories

Many analysis scenarios of trajectories are more concerned with the analysis of the relation between paths than the readability of individual trajectories. Semantic Trajectories are used to determine the number of transitions of players between the different Areas of Interest (AOIs), or establish similarities between paths of different players (Schertler, Kriglstein, and Günter Wallner, 2019). The second case uses a Longest Common Subsequence (LCS) algorithm (Hirschberg, 1977) that works as follows: given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, “abc”, “abg”, “bdf”, “aeg”, ‘”acefg”, .. etc are subsequences of “abcdefg” (LCS, 2011). The AOIs are specified by the user according to analysis purposes and can be subdivisions of the map (Figure 6 left), specific places, or specific objects (Figure 6 right).

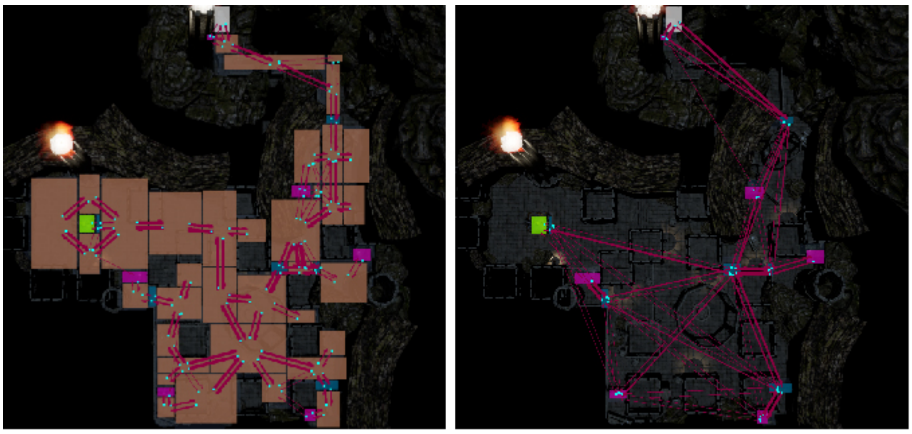


Figure 6: Transition diagrams of aggregated movement between AOIs. Left: Subdividing the map in AOIs. Right: AOIs are objects of interest. (Schertler, Kriglstein, and Günter Wallner, 2019)

Flow graphs

Guenter Wallner, 2018 proposed an algorithm where the map of a game, in this case World of Tanks, is divided in cells and instead of representing each unit’s trajectories they are grouped as semantic trajectories based on similarities for each group. A trajectory is calculated by the overall movement from a place to another, instead of individual trajectories, and groups are presented resembling flow graphs, to give perception about the flow of battles (Figure 7 left). In his new work (Günter Wallner, 2019), after identifying problems with the previous representation that could lead to misinterpretation, the results were substantially improved (Figure 7 right). The white areas are zones of combat and the numbers stand for specific amount of troops. There are two types of arrows where stripes represent troop movement as a group and gradients represent attacks at a distance. The difference between the two representations is that width can represent travel distance of the troops (Figure 7 left) or the amount of troops (Figure 7 right). The latter approach gives a more intuitive representation to analyse how players move through the map, as the stripped arrows have more presence when the number of players is bigger.

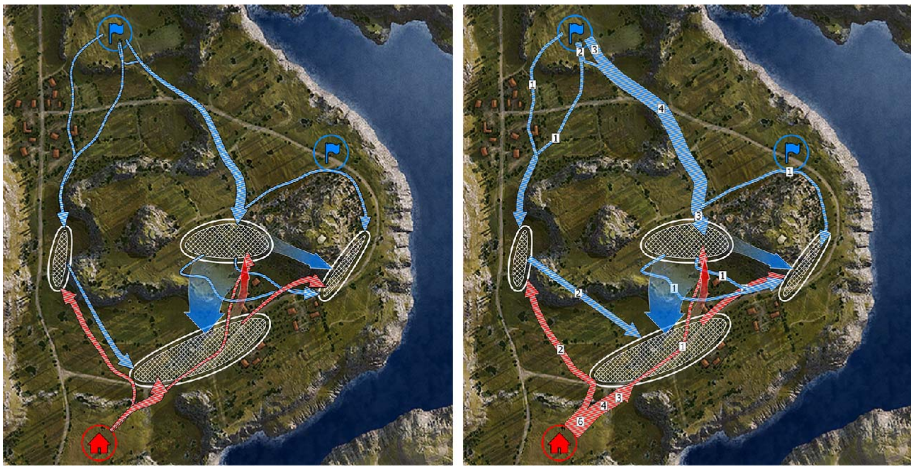


Figure 7: World of Tanks movement with flow graphs (Günter Wallner, 2019). Left: Stripped arrow thickness represents travel distance. Right: Stripped arrow thickness represents number of troops.

Clustering and ST clustering

One approach to solve the overlapping and cluttering problem is to group events with similar attributes or according to neighboring occurrence. Clustering is a process related with machine learning, where given a set of data is possible to classify each element of the set as part of a group(Seif, 2019), (Halkidi, Batistakis, and Vazirgiannis, 2001). The idea is that objects of the same group have similar characteristics while showing a big disparity compared to other groups. There several algorithms available for clustering, each one with different purposes and strengths (scikit-learn, 2019). There are other approaches to aggregate data namely, prediction, change detection, frequent pattern, anomaly detection and relationship (Shi and Pun-Cheng, 2019). Shi and Pun-Cheng, 2019 shows insights about clustering algorithms and specifies different functionalities together with advantages and disadvantages where Density-based spatial clustering of applications with noise

(DBSCAN) seems a common approach. Ordering points to identify the clustering structure (OPTICS) is mentioned as a performance improvement for DBSCAN, but it is also stated that it can not get accurate results so frequently. Jiang et al., 2020 also marks DBSCAN as a good option for clustering regarding spatial feature matching with heavy outliers, as it determines automatically the number of clusters and eliminates outliers, serving as the base for their implementation with its major obstacle being the sensitivity to parameter setting. While the research also references other clustering algorithms, like K-means, they are not considered as they lack the reliability to detect outliers. In this cases, outliers are considered part of the most similar group and a normal cluster, making them less suitable for spatial clustering.

Kanagala and Jaya Rama Krishnaiah, 2016 directly compares K-means, DBSCAN and OPTICS. The first is better when the number of clusters is known and the outliers are less relevant. The others are better when clusters can be arbitrary and the outliers are important to be classified correctly. The last two require two parameters, minimum points to be considered a cluster (minPts) and Epsilon Value (Eps). The minPts controls the minimum size of a cluster and the Eps defines the radius maximum distance to consider a data object as part of the cluster. DBSCAN was considered faster but slightly different parameter settings may lead to different clusters. OPTICS tries to better distinguish closer clusters by maintaining the order in which the data objects are processed. Figure 5 is another example of visualizations with clustering that uses DBSCAN (Günter Wallner, Halabi, and Mirza-Babaei, 2019).

The additional challenge when clustering spatiotemporal data is related with the additional element, time. It can be treated as an attribute but it is not possible to cluster according to it. A common method is to consider time as an alternative dimension like (X,Y,T), where X and Y are coordinates of space and T a timestamp (Shi and Pun-Cheng, 2019). Such as spatial data, in spatiotemporal data clustering it is also important to understand the data and find the best way to resolve outliers, as they can create problems within the visualization (Jiang et al., 2020). To facilitate this problem, such as in spatial clustering, usually density based clustering methods tend to be preferred over the distance based. There are several spatiotemporal data clustering methods namely, progressive clustering, threshold based clustering and pattern discovery (Tork, 2012). In progressive clustering a pre processing of the data according to specific goals comes before the clustering. An example is the work presented by Nanni and Pedreschi, 2006 where the authors propose a data filtering based on specific time periods before the clustering, achieving similar or better results. This also grants performance improvements, since some of the data presented can be unnecessary and is not considered in the clustering phase. A way to make the application of filters viable is through interactivity where the user decides which is the most relevant information from the data set (G. Andrienko and N. Andrienko, 2010).

2.3.3 Visualization techniques used in video games

Nowadays, all virtual games provide several visualization techniques to provide information to players. The approach chosen changes from game to game, but there are always similarities, specially between games of the same genre. In Role-playing Game (RPG) is common to have a big emphasis on health, skills and equipment; in FPS ammunition, the weapon being used and the aim point are essential to keep track of whats happening; and in RTS the map control, the number of troops and the perception of events in the map are the major priority.

Figure 8: Crusader Knights event example (*CruzaderKnights2* 2019)

Crusader Knights 2 is a RTS that presents many events and options to the player. Based on this events, it is possible to inform the actions of a player, when the user hover the mouse over certain areas. Copeland et al., 2018 (Figure 8) shows a big event where the player must make choices and several small events of relevance happening in the map, marking important locations where something worth of player attention happened. The different colors immediately show relations while the position in the map represents the place.

In Figure 9 are shown different methods of visualizations used in a digital version of gaze (Newn et al., 2017). Figure 9 (a) shows a dot to give high spatial precision but can be too intense and has easy over plotting problems. Figure 9 (b) shows a cursor visualization resembling a circle, with the advantage of not overlaying the information below it, making it useful in situations that a marker is needed but what's shown below is also important. Figure 9 (c) is a spotlight visualization and was the approach preferred by the research participants, for its precision and yet subtlety that comes with the surrounding blur. The idea behind the Figure 9 (d) is a fixation visualization to give a notion of where the player went using the arrows to point the direction. Figure 9 (e) shows a scanpath with points of focus related between them, if a player stays more time in a place, the dot becomes bigger creating a notion of motion. Figure 9 (f) shows a fixation trail with the length, duration and direction, of a player where the opacity and circles distance can be related to specific attributes. Figure 9 (g) is the typical heatmap, where warm colors represent spots where the player had more focus and cold colors represent spots with less player focus. In Figure 9 (h) is a convex hull visualization representing an area of interest that stretches according to the player interest in specific events or places is displayed by a low opacity shape. Finally, Figure 9 (i) is a bee swarm which consists in a mix between the convex hull and points, where the blue shape shows an area of interest observed and the dots represent specific events.

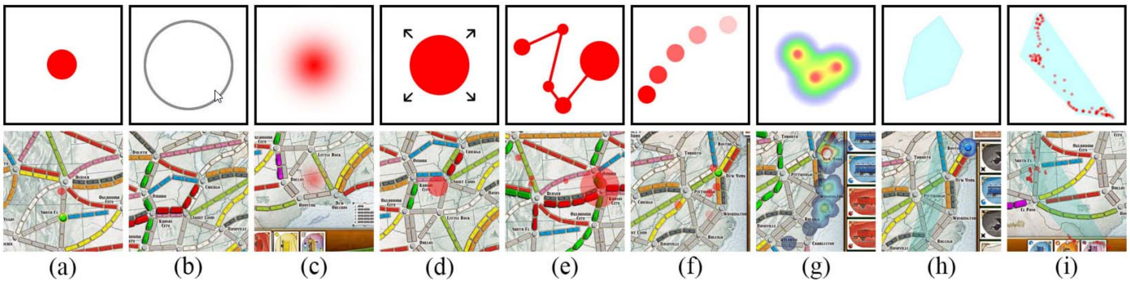


Figure 9: Different visualization techniques (Newn et al., 2017).

2.4 Summary

Related work shows the importance of eSports as a professional career and the proeminent future of virtual games as an entertainment and career. Visualization techniques are present in everyday human life and are an approach to efficiently analyse data as this can become voluminous due to telemetry data gathered from video games.

To analyse these complex data sets usually, including ST game related data, is common to use visualizations to discover and extract behaviour patterns and strategies. Several visualization techniques to analyse ST data have been proposed and in particular in previous work, the animated map was explored to analyse ST data associated with a LoL match for a single player (Moucho, 2018). However, visualizations of large data sets in multiple games analysis can cause visual problems, like cluttering and overlapping of the information. To solve this problem it is crucial to explore approaches like aggregation, clustering or filtering.

To achieve this, Visualeague III, presented in Chapter 4, uses filtering and visualizations together with a common clustering algorithm for ST data, DBSCAN, due to its strong outlier detection, not requiring a predefined clustering amount and good results in performance.

Since knowledge of the case study is mandatory to understand what kind of data is more relevant to extract information, an introduction to LoL and its competitive aspects is presented in the next chapter. Additionally, the previous prototype, VisuaLeague II, is also described.

Chapter 3

League of Legends and VisuaLeague II

This chapter presents an overview of the LoL game and the previous prototype, VisuaLeague II. Section 3.1 explains how the game works with emphasis on the most common match type. Section 3.2 describes the competitive aspects of the game. Section 3.3 shows common web applications used by players and coaches to analyse LoL matches. Section 3.4 describes coach methodologies to analyse professional training matches, taking into account an informal interview with a professional coach. Finally, Section 3.5 presents an overview of the previous prototype, VisuaLeague II, that focuses on the analysis for a single player in a single match. Additionally, this section also describes the technologies used and champion paths processing.

3.1 League of Legends

LoL (Riot Games, 2019a) is a fast-paced, competitive online game that blends the speed and intensity of an RTS with RPG elements common in the MOBA genre. It involves two teams of five players competing with each other to defeat the opponent's base (Nexus). Each player, referred to as a summoner, can choose a character (champion), from a roster with already over one hundred possibilities and continuously increasing, each one with different strengths and weaknesses. The game offers different game modes and maps. Summoner's Rift is the standard moba mode (and the name of its map) and the only one featured in professional level play. A match takes place in a squared map composed of three lanes, top, middle, and bottom that connect the player's base to the enemy's (Figure 10). To win a game, the player needs to push down their lane into the enemy base and destroy the main structure (Nexus) at the center of their base. Between the lanes lies the jungle, an area filled with neutral monsters (mobs). These monsters will not attack the enemy base, but they can be defeated for additional gold and the champion gains temporary bonuses (e.g, Dragons and Baron). Each player can use a unique champion to fulfill a specific role in the team: one player goes to the top lane (Top) and another to the mid (Mid), two players go to the bottom lane (Attack Damage Carry (ADC) and Support)

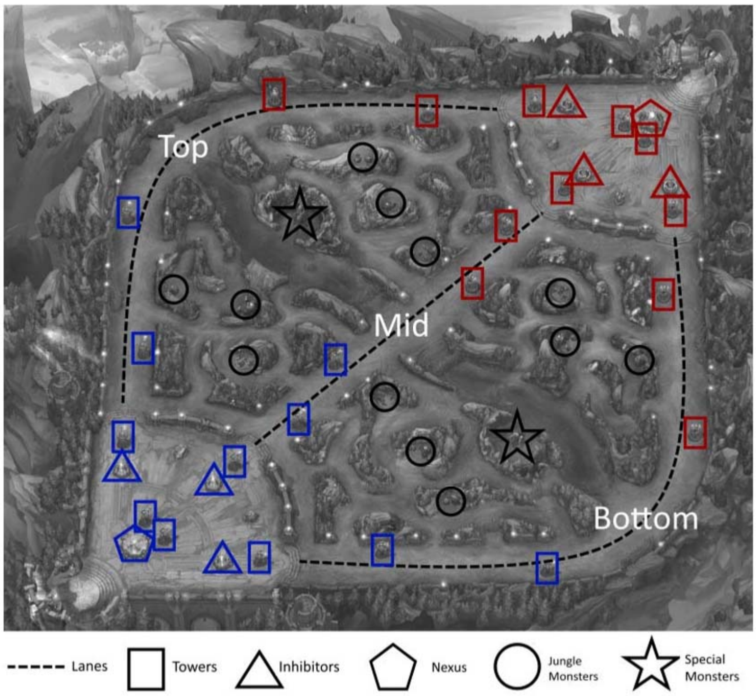


Figure 10: League of Legends map with important locations (Gonçalves et al., 2018)

and the remaining player goes to the jungle (Jungler). However, this is the usual method and different strategies can result in a change of roles.

Periodically, AI-controlled soldiers (minions) spawn at the Nexus and march down each lane towards the enemy base, attacking enemies they encounter along the way. When minions die, they give champions experience points (XP).

Towers are defensive structures that protect each lane at specific intervals, hitting enemies that come within range. Usually tower prioritize minions, but if a player attacks another player s/he will be the target instead. Some other structures, known as inhibitors, are located where each lane meets the base on both sides of the map.

Players see the full map, but only have updated vision in locations with an allied minion, structure, player, or some special ability, item or a ward is used.

Normally, a match's gameplay can be divided into three stages: early-game (until 15 minutes), where players mostly stick to their lanes; mid-game (until 25 minutes), where the team starts grouping to help push a lane and helping allies by invading enemy lanes; and late-game (until the end), where team fights are predominant, trying to invade the enemy base. Summoner's Rift matches typically last between 30 and 40 minutes until a Nexus is destroyed. The specific time for each phase can change between version of the game, but the three game stages are always present.

Before starting a match, each player can choose a champion, representing a different playstyle and experience. The player can also choose a group of optional features, named build. Build options are attributes with additional special effects (runes) and two special abilities (summoner spells) that are independent of the selected champion and can have major impact in the play style and strategy of a team as well. During this process, champions are selected in groups that alternate between teams. One player from a team goes first, then one player from the other side. After the first player selects a champion (first pick) the remaining summoners are grouped in pairs and alternate in the same way. Between this selections, both teams can choose champions to exclude form the match (bans). Before the first pick, three champions are selected to ban and after each of the upcoming groups select their champions, is available one ban totalizing ten champions banned in a match (five for each side, one for each player).

3.2 Competitive environment

There are two different concepts when referring to the competitive environment in League of Legends. The first one, described in Section 3.2.1, is accessible for everyone, and usually referred to as ”solo queue”, which stands for games a player does for ranked matches in Summoner’s Rift. This ranking system shows player competitive level as an individual. The other side represents the professional teams that compete in tournaments for eSports and is discussed in Section 3.2.2. It is important to distinguish this two concepts as they require different approaches to strategies and player behaviour, and influence the corresponding analysis.

3.2.1 Solo queue

There are a total of thirty six available leagues to place a player, which represent their quality in game compared to others of the same region. This only applies to people who play ranked matches and want to take part in a competitive side of the game. Otherwise, players will be assigned with an unranked placeholder.

The player rank is determined by the number of victory games over defeats. Every time a ranked game is won the player gets additional League points (LP) and when the game is lost the opposite happens. To go up in leagues, one hundred points are required for each one and during the transition, a trial of five matches occurs, where the player must win in three of the five chances. LoL also tracks a numeric value for each player named Match Making Rating (MMR) which represents the player quality. This value is not showed in the game but it considers the average performance of the player, determining if s/he is in the correct league. In case the player’s MMR is above average in the current league, s/he will gain LP much faster. The reverse idea also applies, if the player MMR is lower than average in the current league s/he will need more games to pass to the next league and is more likely to fall in rank.

Web applications for game analysis only offer match data from this type of competitive environment (detailed in Section 3.3).

3.2.2 Professional teams

Professional teams playing in eSports at a competitive level are often a group of five players with high solo queue rankings and a variant amount of substitutes that practice together in custom games (scrims). This game mode is distinct from the ranked and normal games as it is created by an user with availability restrictions, and only players that participated in that specific match have access to it.

Teams' constitution constantly fluctuates and members are frequently changing, namely players and back stage personnel, like coaches and analysts, specially in smaller teams. This situation creates a problem when searching for data sources with team information. Other factor to consider is that a professional player can have multiple accounts, and usually the summoner name of those accounts is not the same as the one used in tournaments.

Riot's API itself does not keep track of this particular information for professional players and teams. To solve this limitations, usually coaches take their own annotations during the matches for future analysis. There are web applications that focus specifically in tracking professional player accounts like lolpros.gg (lolpros, 2020).

3.3 Web applications for game analysis

As one of the major games on eSports and by having a MOBA game-style reliant on well timed strategies and decisions, is important to have tools that help to optimise player decision making. Some of the most well known applications were mentioned by one of the two interviewed professional coaches, and will be discussed in this section as they are used for every type of player and analyst. From a casual player perspective it is important to know the trending and try to mesmerise builds and strategies to get better at the game. A requirement for professional players is that applications provide more detailed information about themselves, the team or their adversaries. Couches, analysts and developers also use these applications to keep track of player performance and find instabilities in game's balance. Some examples are OP.GG, Probuilds and Mobalytics.

3.3.1 OP.GG

OP.GG (*Lol Tier List* 2019) is one of the most used web applications to analyse data of a specific summoner. In Figure 11 (left) is an overview of the recent match history of a player. At the top, OP.GG shows the summoner name and icon for fast identification, and right above it, shows all the player's journey since s/he first started playing in older seasons and the corresponding rank for each one. This way it is possible to have a notion if the player is a recent player or an older one together with performance in each season. Bellow the summoner name is the most recent information: on the left, an overview of player's current rank and the recent champion picks with the respective win rate and Kills, Deaths and Assists (KDA) which can then be expanded in a more detailed view in the tab ”Champions”. At the right, it presents an overview of the recent matches. The user can choose a specific match to get details of champions, runes and builds. For a more detailed analyses, it is possible to move between tabs inside each match view (Figure 11 (right)), and access to detailed match analyses, event logs, in-depth build previews, bar charts and line charts comparing each team. It is also possible to compare each champion performance with one champion of the enemy team.

OP.GG is often used for its statistical data for a single player in solo queue matches, making it a very efficient tool to find specific player strengths and weaknesses. However, when analysing multiple players it becomes less suitable, as the builds displayed, only refer to the selected player. Also spatiotemporal data is barely present with the only option being a kill map, where the locations of kills are marked. The information for custom games data is also not available.

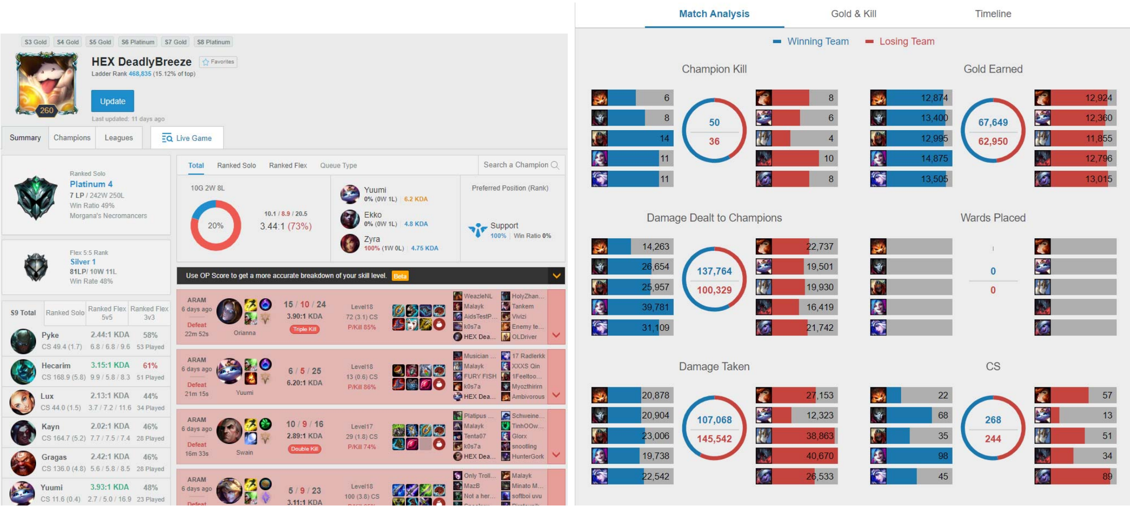


Figure 11: OP.GG visualizations examples (*Lol Tier List* 2019). Left: profile overview. Right: analysis example.

3.3.2 Probuilds

Probuilds (Probuilds, 2019) is an application focused on the champion information rather than a summoner itself, that gives an updated overview of popular players’ most recent choices to approach each champion gameplay. This site is considered one of the best to keep in perspective what is currently happening in the game from a competitive perspective as it offers a list of recent matches, champions with the most winning ratio, champions most played and high rank players’ accounts. It gives the option to search for a specific champion or a specific player from the available list, in order to see recent matches in a more detailed way, as shown in Figure 12 (left).

An example of information obtained from Probuilds is down, in Figure 12 (right), namely the timestamp for a certain acquired item. This is very useful for professionals to analyse the game course and also for casual players to know that at a certain time, they are suppose to have a certain item and try to achieve this objective.

Statistical data is once again the major strength of this web application, with a focus on champion builds and providing direct comparisons between different options. Although build orders and specific timestamps to obtain an item can not be directly compared, as it is necessary to open a new tab to access that information. Probuilds does not have access to every player account in the game, making impossible searches for some players. It also provides extensive analysis on player builds through sorting tables and timelines. But, it lacks on player focused analysis, presenting only very generalised statistical data for players matches and player win ratio for the top ten accounts.

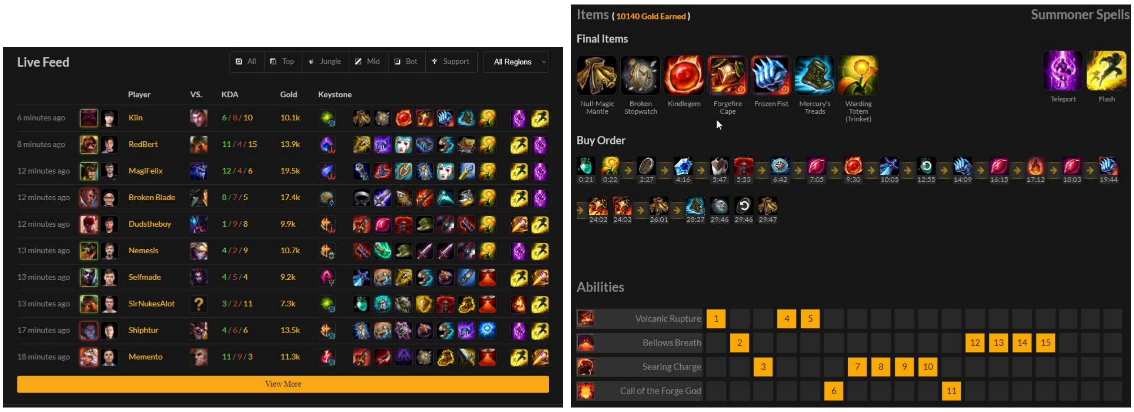


Figure 12: Examples from Probuilds website (Probuilds, 2019). Left: General view of recent matches. Right: Example of an item build order.

3.3.3 Mobalytics

Mobalytics (Mobalytics, 2019) is the web application, currently in beta version, with more options to customize analysis. It offers a wide variety of data, including full descriptions and some insights of professional players in descriptive texts about their history, with the champion and the game. It gives access to a tier list, where the potential of each champion is rated according to the current patch and has various tools to analyse matches. One particularly interesting is the ability to predict the game of a player that gives the best strategy to follow in that match, acting as a virtual coach. Another feature that captivates interest is the progress analyses feature, that dives into more specific analyses of the player (Figure 13 (left)). Although this feature requires a subscription, a video demonstrating its potential can be found in the web page. Mobalytics also offers the ability to customize the dashboard with the topics s/he considers most relevant (Figure 13 (right)).

The most common visualization techniques are histograms, charts and radars used in individual statistics and aggregations for matches over time. Mobalitics does not offer access to spatiotemporal data nor access to custom games data.

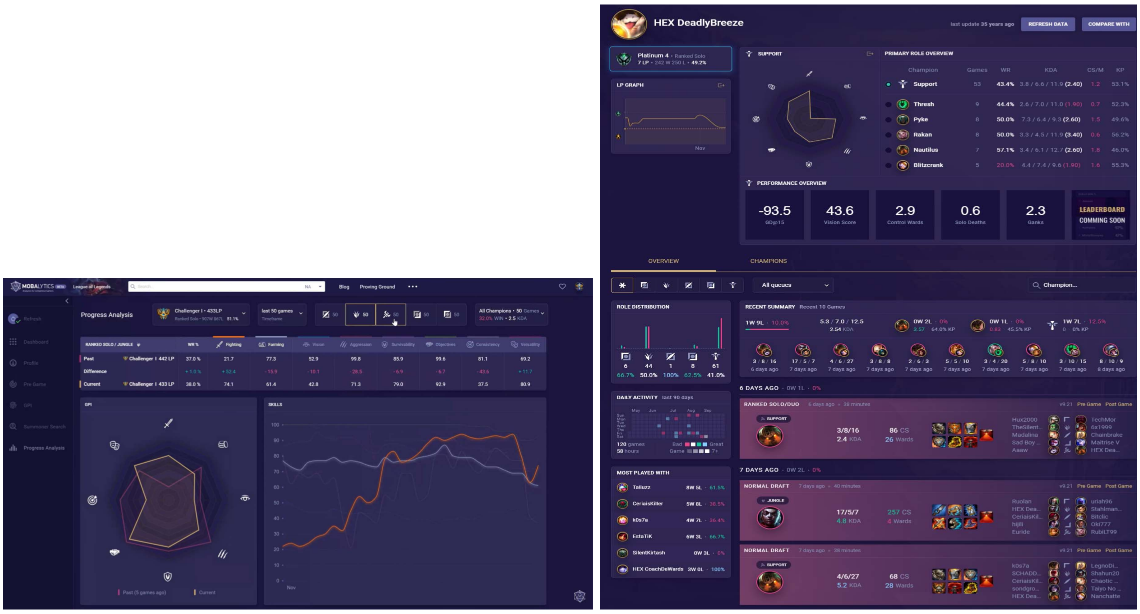


Figure 13: Mobalytics visualizations examples. Left: Progress analyses. Right: Personalised user dashboard (Mobalytics, 2019).

3.3.4 Other applications

The tools mentioned above are only a fraction of what is currently available for the analysis of League of Legends. Other examples are ”gol.gg” (gol.gg, 2019), ”lolcounter” (lolcounter, 2019), ”championGG” (champion.gg, 2019), and ”U.GG” (u.gg, 2019). These tools are used by players and analysts with less predominance as they add very few extras to the ones already described. Mobafire (*MOBAFire* 2019) is another tool that serves as a tutorial. It is a community driven website, where users can post builds of a champion to others test and rate. It is mostly redirected to new players who want a explanation about what the champion does and what to do with it in order to succeed. It does not offer the analysis potential as the others mentioned above, but it is a different approach to show worthwhile trends about the game.

3.4 Coach analysis methodologies

The development of this project had a very close support of a League of Legends professional team coach, who showed and explained part of the process used to keep track of his team performance and more recent games results.

To train teams and test players, coaches and team managers organize events where two teams compete one against another, called scrims which are basically custom games, as stated in Section 3.2.2, with players that look promising from their solo queue rank (Section 3.2.1), and want to take part in a professional format of the game. From the several ways of collecting player data (Section 3.3), OP.GG tends to be the predominant tool, due to its intuitiveness and versatile options to search different information solo queue related, and does not require payments to access the data.

Coaches usually follow the scrim, while it is being played and take notes of results and major happenings to an excel stylesheet, in order to later revisit the game and find ways to overcome difficulties and reinforce strategies. The notes taken are very summarized, for instance, ”victory, hard game”, ”victory, easy game”, ”defeat” are ways to determine which games were more important to revisit due to how hard it was to win or if they could not win at all. Besides that, some specific time periods are noted if something of importance happens that was possibly a key factor to the game's result. For example, a missed objective or a wrong start of a team fight.

Besides the information they get from the scrims, coaches have the job to search for their players' solo queue information and create strategies to strengthen their gameplay. Information like most played champions, position/lane, item builds and win rate is something a coach is always looking for. One example that the coach gave is the analysis carried out from player historic matches, if the lane, or champion, that a specific player uses more often is what s/he should be using considering win ratios. A more practical example is when a player is playing top, in several games, with a certain champion and often loses and has a position with higher win ratio but less matches. In this situation, the coach will ask for the player to try that second position instead.

3.5 VisuaLeague II

This section describes the previous prototype, VisuaLeague II. Section 3.5.1 presents technologies used by VisuaLeague II and explains in depth the approach to simulate champion movement in the animated map visualization. This technologies are further developed in Section 4.4 explaining exactly their function in the new version, VisuaLeague III. Section 3.5.2 gives an overview of VisuaLeague II.

3.5.1 Technologies and champion path processing

VisuaLeague II uses technologies which contributed to overcome limitations of the previous work. Node.js (*node.js* 2019) and Express (*express.js* 2019) are responsible to create and organize the web application in a Model-View-Controller (MVC) architecture. Materialize (*Materialize* 2019) was used to facilitate the interface design for the application. Leaflet (*Leaflet* 2019), noUISlider (*noUiSlider* 2019) and ChartJS (*ChartJs* 2019) were responsible for the implementation of the interactive map, timeline and charts visualizations, respectively. CouchDB (*CouchDB* 2019) is used to store data extracted from Riot API in order to optimize requests and Neo4j (*Neo4j* 2019) is a graph database used in the animated map visualization to simulate the available paths a player can take.

Figure 14 left shows an abstract representation view with the created nodes and how they are connected in Neo4j. In total there are 129 nodes and 211 relationships. Figure 14 right shows the nodes placed in the Summoner's Rift map with the representation of the connect location of nodes and possible paths, avoiding the walls.

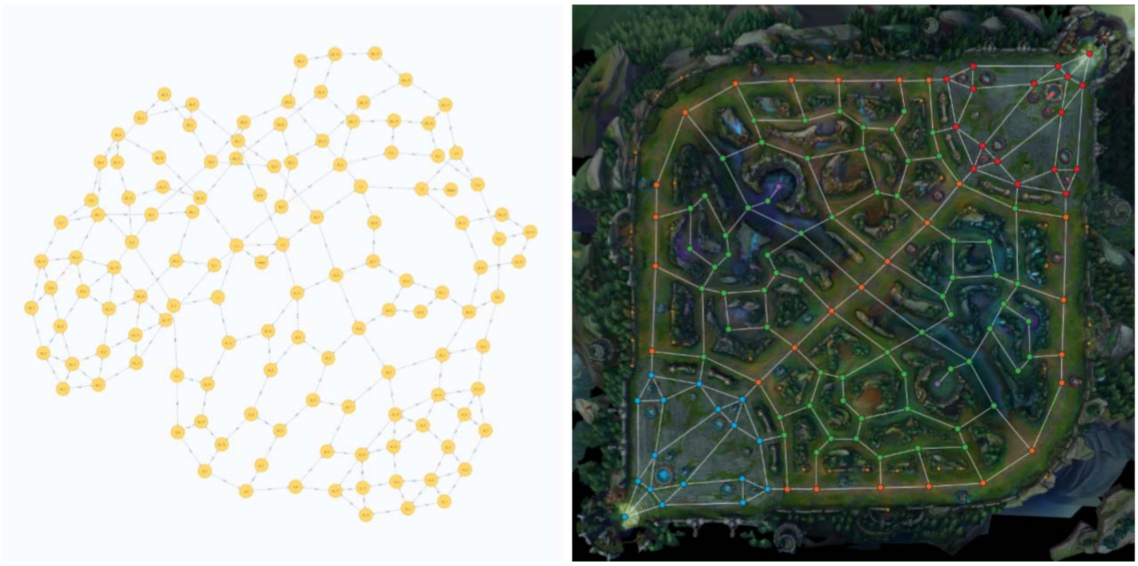


Figure 14: Neo4j nodes (Moucho, 2018). Left: Abstract node representation from Neo4j. Right: Nodes representation in LoL's map.

When loading the animated map visualization for a certain match, Neo4j (*Neo4j* 2019) is requested to create paths for all players. Each node is connected to at least another node by a relationship named path. This relationship has a distance attribute, resulting from the distance between two nodes. Two plugins were used to make the path closest to its actual representation, Neo4j Spatial and APOC. The Spatial library enables the use of spatial operations on data, like searching for data within a region or within a specified distance of a point. The first step was to create a point layer where later the nodes will be referenced to. Using the Cypher language, the nodes were created with unique id's for better readability, and more two parameters representing both latitude and longitude, required by the library. After being created, they are added to the point layer and the available spatial functions can be used. There was only one function used in this case, the closest function, which finds all nodes in the layer within the distance to the given coordinate. This function is important to know what is the first node the player must go to. The APOC library consists of many procedures to help in areas like data integration, graph algorithms or data conversion. With this library, the Dijkstra's algorithm was used to find the shortest path between nodes based on their distance.

With the player's coordinates at a timestamp, we first find the closest node to his/her coordinates. The coordinates of the following timestamp are also used to find the closest node, establishing the path's start and end nodes. Making a call to Neo4j with these nodes using the Dijkstra's function, it will return the shortest path between them, with information of every node the player must move to in succession, avoiding going through walls.

With the paths created, the next step was to calculate the movement's duration between each nodes. To find speed, we need to divide distance by time. Dijkstra's function not only returns the path, but also the total distance. This total distance is incomplete, though. We also have to consider the distance between the point where the champion starts and goes to the first node, and the distance between the final node and the point where the champion must be. So, dividing total distance by the subtract between the end timestamp and start timestamp, we get the constant speed a champion must travel through the route, mimicking LoL's gameplay.

Having the speed constant, we loop through each route's node. To calculate the duration between the current node and the next, we need to divide the distance by speed, adding it to the duration's array, not forgetting the first and last points that need to be considered. Finally, the route must be added to the path array. Both arrays will then be used by the MovingMarkers plugin, simulating movement. There is the possibility that the start and end nodes are the same, which will make the Dijkstra's function to not return a route. The path will then be directly between the start and end nodes, and the duration will be the subtract between the start and end timestamps. Hypertext Transfer Protocol (HTTP) requests are being made to Neo4j to create the paths for all players when a match is being loaded.

3.5.2 Prototype functionalities

Prior to the main visualization of the analysis tool is a front page, Figure 15, where the user introduces a summoner name and the respective region to search for matches in couchDB or in Riot's API. After this, the user choose in a list (recent matches with a short overview of player statistics, game type and result) the match s/he wants to analyse in a new view of VisuaLeague II (Figure 16).

VisuaLeague II uses an animated map to show the position of every player, every minute, and other ST events that are crucial to player analysis. On the right of the Figure 16, the information panel presents information about players' statistics (eg. roles, scores, gold). The information is updated as the game progresses. VisuaLeague II is also composed with a minimap, timeline and filters to offer users context of the surroundings when they zoom on the main map; control the animation; and filter the information displayed, respectively.

3.6 Summary

League of Legends is a game where two teams fight to destroy one another. Its one of the most popular eSports' game genre and tends to be highly competitive, making analysis

Figure 15: VisuaLeague II match search page (Moucho, 2018).

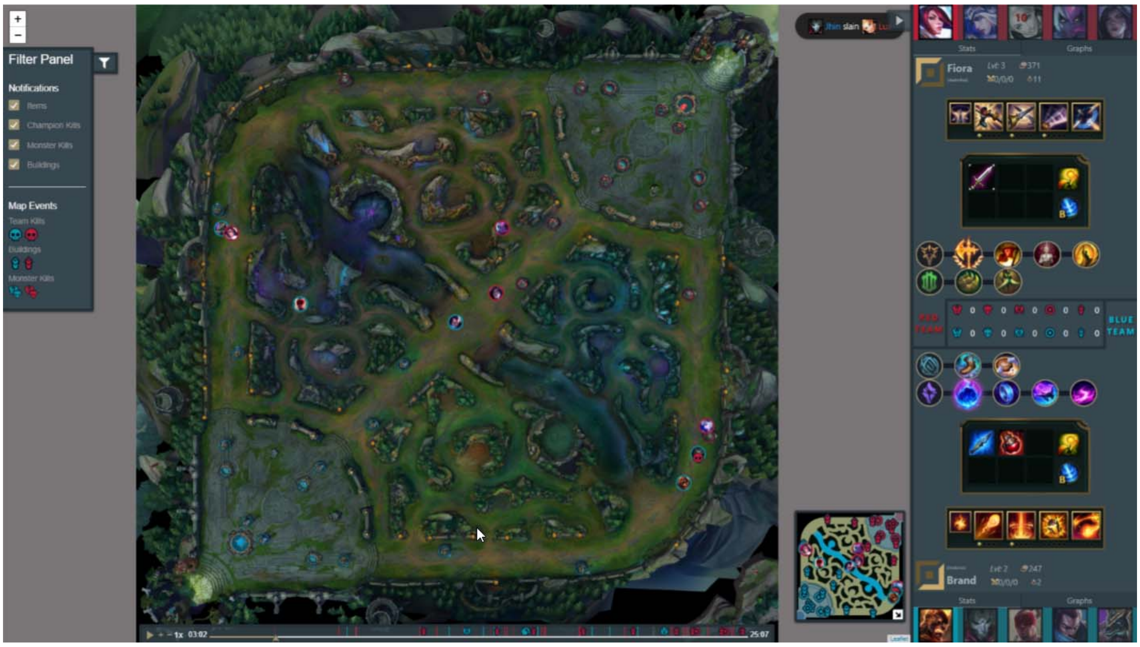


Figure 16: VisuaLeague II interface (Moucho, 2018).

tools popular for professional players, coaches, analysts and casual players. LoL is played in five man groups, but professional teams an players can be very ambiguous as there is a mix between what is the solo queue with in-game ratings and competitive eSports tournaments. LoL is the case study due to its popularity and because it offers free access to Riot API. There are already efficient web applications to study player performance for LoL with the main focus on statistical information, like KDA, rank info, items and champion picks. Although they do not have methods to analyse ST data, the main objective of VisuaLeague is to provide visualizations to ST data, among others.

A common way for coaches to train professional teams is through scrims, considered as custom games by Riot. These are the game modes where happens professional training matches. This makes them an important factor to consider in analysis but, due to its absence in Riot API match list, current analysis tools do not offer ways to access that data.

VisuaLeague II is already an efficient tool for analysing League of Legends' games. Relying on an animated map to showcase player positions over time, a timeline bar to navigate or watch event occurrences and ways of displaying events or static data like notifications, graphs, and tool-tips. This approach proved to be a good solution for the analysis of a single match from a single player. However, it becomes less efficient when the focus is the performance of a player or team over multiple games, since the user would have to open multiple maps and analyse each match individually.

The challenge to VisuaLeague III is to take advantage of the animated map visualization, re-implement it to be compatible with new data from Riot and expand their functionalities to multiple games and team analysis.

The next chapter will present the analysis and design of VisuaLeague III.

Chapter 4

VisuaLeague III — Analysis and design

VisuaLeague III takes the structure of its predecessor expanding the range of possibilities in multiple types of analysis to better fit the needs of coaches, players and analysts by adapting old features like the animated map and bringing new ones as wider range of match types, aggregations, filters, team search and multiple matches.

The first part of the project went through an extensive study of VisuaLeague II and its approach to retrieve data from Riot API in order to update obsolete methods and prepare new ones. In previous experiments were already identified the information that players consider more relevant and how they use it to analyse their performance. In VisuaLeague III, the intention is to capture the requirements of coaches. In order to achieve this, two professional coaches were interviewed, Section 4.1. Section 4.2 presents a brief description of the developed prototype. Section 4.3 shows prototype's architecture, explaining the functions of each module. Adaptations discussed next in Section 4.4 refer to the technologies described above in Section 3.5.1 mentioning important changes. Section 4.5 explains how the data required is extracted from the APIs and Section 4.6 shows how it is then processed by VisuaLeague III. Section 4.7 is where Design solutions are presented for each different type of analysis VisuaLeague III can offer.

4.1 Professional coaches interview

Before starting the development of VisuaLeague III an interview with two professional LoL coaches took place in order to understand their needs and analysis methodologies. The document that emerged from the meeting can be fount in the Appendix A. With this interview was possible to understand that coaches use a wide range of tools, namely OP.GG (OPGG, 2019), Probuilds (Probuilds, 2019) and Mobalytics (Mobalytics, 2019) to know performance of individual professional players watching the most played champions, champions with more win rate, builds, summoner spells, KDA and solo queue rank. Also they use databases to keep track of players' alternate accounts, as they're always changing, with the most popular web application in this regard being lolpros (lolpros, 2020). Gamepedia (gamepedia, 2020) is usually used to search information about players' identities and professional career. When analysing a game, one big concern is to know if the player buys the right item at the right time and if they all go to the base at similar times. The game usually is not all relevant, being only specific time frames important. In a further discussion with one of the coaches, he reinforced that the most important time frames would be every five minutes and at eight minutes, where the herald spawns. Ward placement is very relevant as well, it can give very useful information to the analysts, however Riot API doesn't give enough data on this matter. They also mentioned the difference between solo queue and professional games, emphasizing that the games where professional players practice are custom games and cannot be accessed like normal games.

To understand the requirements in terms of multiple games, questions were asked about which type of analysis they would do for multiple matches and what type of aggregated information they would like to explore related with this matter. They didn't seem used to analyse that many games together for professional player training, the reason being constant changes of the meta and game strategies. On average they usually analyse something around five games, being, in very rare cases, a maximum amount of twenty for scrim matches or custom games. However, in solo queue the number of games relevant for analysis goes higher with twenty to forty games being a good average range to consider. After some thoughts they considered the most relevant aspect in multiple games would be movement, wards, bans, and events like turrets destruction.

4.2 Prototype description

VisuaLeague III is a web application where a user can visualize information related to matches that took place, independently of whom participated in it. The main goal is to assist players, coaches and analysts in player performance analysis for LoL matches with focus on coaches and the professional environment of the game. LoL is very popular worldwide and offers a voluminous amount of data to analyse, making it a suitable candidate for study and research.

With the previous prototype limitations (Section 3.5) there was a need to include new analysis methods for one player, teams and professional training sessions. For each of those is possible to focus in analysis of one or multiple matches which are loaded according to user's needs with the help of filters.

When accessing VisuaLeague III, the initial screen presents the analysis for a single match from a single player (Figure 17). It provides a top navigation bar where is possible to navigate through the different analysis methods available or search for a player match list in a world region.

After the selection, it is presented information about the player account (name, level, icon and rank information) and each match (result, scores, game mode and dates). For every match is possible to load a new page with detailed information about the game chosen and its participants, through the animated map visualization referred in Section 3.5.2 and further developed in Section 4.7.2.

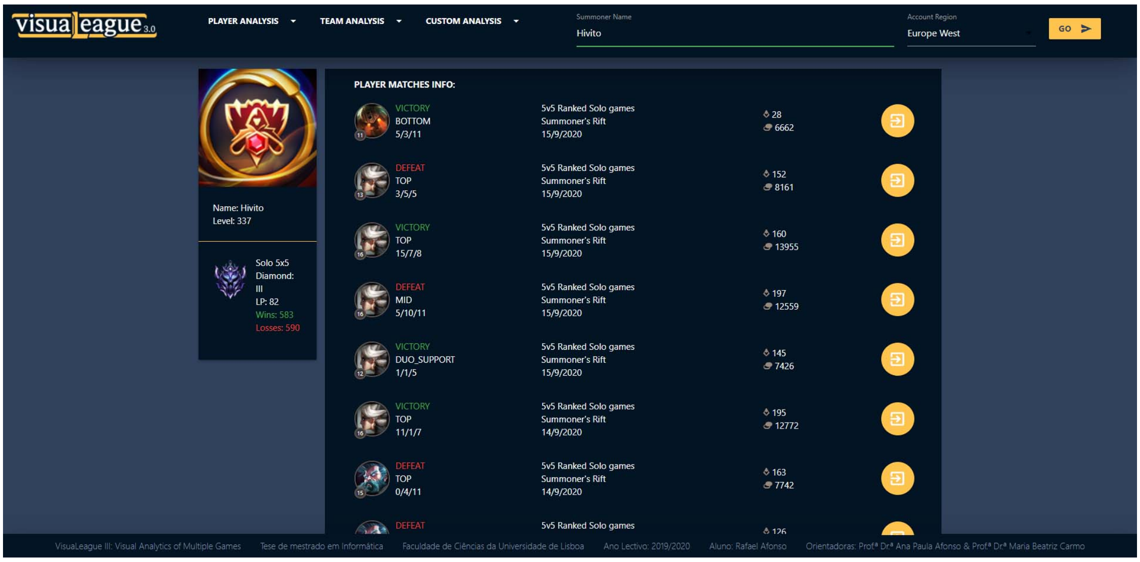


Figure 17: Player analysis for single matches. Searched player recent games display.

Figure 18: Player analysis for multiple games (eleven games). Left: Maps with kill events. Right: tower destruction events.

Figure 18 represents an example of analysis for multiple games of a single player search. In the selected tab are available visualizations for spatiotemporal data, aggregated from the matches loaded. Here the user also searches for the wanted player through summoner name and region, but before loading the match, filters will be requested to customize the type of game, number of games, and game modes. Currently, in Figure

18, are displayed kill events at the left map and tower destruction events at the right. This visualizations can be customized by the user to show other information like player movement or elite monster kills. It is possible to filter data regarding important aspects of the matches, like team side (red and blue), for instance. It is also possible to use a timeline to analyse a specific time frame only and group positions through clustering.

This and the remaining methods available are further described in Section 4.7. All of the design and user experience decisions were based on iterative and informal interviews with coaches. The architecture, technologies, data gathering, data processing and design features are discussed in the following sections.

4.3 Architecture

This Section explains the structure of the application developed enumerating the major parts: views, controllers, models, databases and APIs. Views are the front pages with which the user will interact, controllers are responsible for handling requests made in the views and passing them to the corresponding models so these can process the data and respond. Databases are VisuaLeague's repositories to store required preprocessed data. Finally APIs are data sources where information relative to players, teams and games is gathered. Figure 19 shows the MVC architecture where is possible to see interactions between the components.

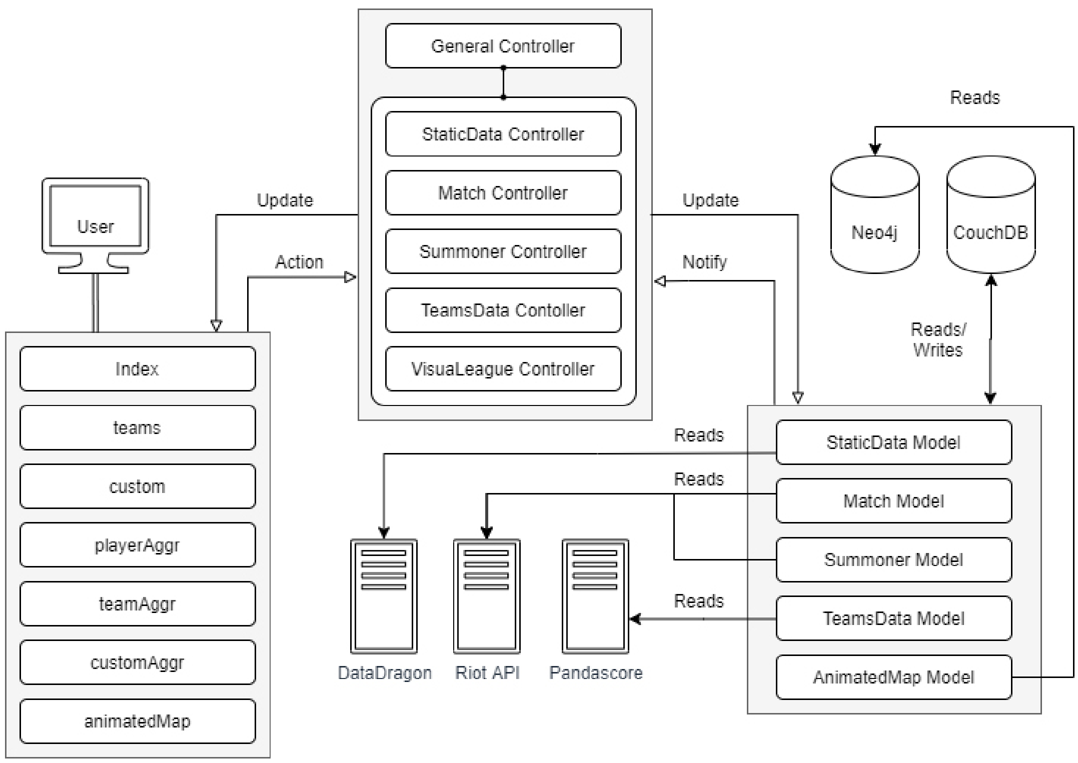


Figure 19: VisuaLeague III architecture

Views

Index — Responsible for the initial front page where users can access player matches for one player.
Teams — Responsible for teams page where users can access matches for a player, groups or professional teams.
Custom — Responsible for custom games page where users can access custom games and scrim matches.
PlayerAggr — Responsible for visualizations of multiple games for one player.
TeamAggr — Responsible for visualizations of multiple games for a player, groups or professional teams.
CustomAggr — Responsible for visualizations of multiple games for custom games and scrim matches.
AnimatedMap — Responsible for the animated map visualization.

Controllers

General Controller — Commands all the other defined controllers and responds to the views.
The remaining controllers — Communicate with their respective models, passing that information to the General Controller.

Model section

StaticData Model — Champions, items, maps, constants, runes lists and summoner spells are all part of the static data. These are referred as static because they are rarely updated and are storen in DataDragon.
Match Model — All data regarding matches is handled here. This model gathers content from Riot's API as its functions are to preprocess data relative to a match or a group of matches depending on request. In the case of more complex data gathering as in team analysis it is also responsible for filtering matches from players' group match lists. Some of the content is persisted in CouchDB, spare request to APIs.
Summoner Model — Fetching League of Legends summoner names is done by this model. The search is made with the summoner's name and region since it is the common sense among players, but it could also be used a summoner's id or the account id directly. The request's result is persisted in CouchDB in the case of single player analysis. Team analysis uses this model to verify user identities only and custom games analysis doesn't need this model at all. Ranked information is also handled here by making a specific request for player's ranked matches statistics and league to Riot API when needed.
Teams Model — Responsible for creating and retrieving teams from couchDB made by the user and loaded from Pandascore.

AnimatedMap Model — Model responsible to preprocess data for the animated map visualization. The routing system, duration, stats over time and events are all processed here. After everything is dealt with, the content is saved in JSON, ready to be used by the front-end. Neo4j only communicates with this model as it is where the routing is formed.

Databases

CouchDB — Where is stored Summoner, teams and matches data in JSON format.
Neo4j — Graphical database where are stored the nodes and its relationships to create routes for champions' movement in the animated map Visualization.

APIs

DataDragon — Where all static data is gathered from.
Riot API — API responsible to provide all data for summoners and matches of players.
Pandascore — API responsible to provide the professional teams and players' data.

4.4 Technologies

This section presents the main technologies used for VisuaLeague III together with adaptations from the previous prototype for each one. Section 4.4.1 shows Node.js (*node.js* 2019) and Express (*express.js* 2019), responsible for the creation and organizations of the web application in a MVC architecture. CouchDB (*CouchDB* 2019), Section 4.4.2, is used to store data retrieved from Riot API. Neo4j (*Neo4j* 2019), Section 4.4.3, is a graph database used for pre process player paths. Materialize (*Materialize* 2019), Section 4.4.4, was used to facilitate the design of the application. Leaflet (*Leaflet* 2019), noUISlider (*noUiSlider* 2019) and ChartJS (*ChartJs* 2019), Section 4.4.5, were responsible for the interactive maps, sliders and charts, respectively, used throughout the prototype to create visualizations and user interaction with filters. jDBSCAN (*jDBSCAN* 2020), Section 4.4.6, is an adaptation of the algorithm DBSCAN for Java Script (JS) used to cluster spatiotemporal data,

4.4.1 Node.js and Express

Node.js (*node.js* 2019) is the environment to create applications in JavaScript that was elected as the server run time. It is designed to run directly in the machine, be it a computer or a server, omitting JavaScript APIs specific for browsers and adding more traditional ones of Operating System (OS), supporting HTTP and files. It also supports a wide variety of libraries that will be of use.

Node.js has a great performance as well as scalability and the event-driven architecture capable of asynchronous Input/Output (I/O) makes it a good choice for a web application.

Express (*express.js* 2019) is a lightweight framework for Node.js that helps organize a web application into a MVC architecture on the back-end. It manages routes, requests and views (ExpressNode, 2019).

To adapt Node.js and Express to VisuaLeague III, major changes came when adding new models and dealing with the files for multiple games for multiple players, Section 4.7.6, that became more demanding, making it necessary to adjust this parameters.

4.4.2 couchDB

CouchDB (*CouchDB* 2019) was the database selected to save data from players, sparing Riot API from handling so many requests which could become very demanding and also improving the search speed of VisuaLeague. Another advantage is that, if for some reason the servers of Riot go down VisuaLeague still has access to the last data, although it cannot check for updates. CouchDB was the database selected because it speaks JSON natively and saves information without a specific structure.

To adapt couchDB to VisuaLeague III was necessary to rebuild the database and update methods to communicate with it due to versions compatibility. New adjusts to file sizes was necessary to make storage of multiple games for multiple players possible. The interaction of the database with the application will be further developed in Section 4.5 and Section 4.6.

4.4.3 Neo4j

Neo4j (*Neo4j* 2019) is an intuitive graph database implemented in Java that uses Cipher Query Language through a transactional HTTP endpoint. Its visual representations makes it easy to understand and visually relate nodes which can have attributes and relations between them enabling the ability to request paths and specific nodes by queries. It also has a wide variety of plugins including functions to determine the shortest path between nodes which will be of use to implement the pathing for the players in the map. The two plugins used were Neo4j Spatial and Awesome Procedures On Cypher (APOC), the first serves to search data by spatial operation like within a region or a specific distance to a point. The closest function is how the next node a player goes to is searched, by finding all nodes in the layer within the distance to a specific coordinate. The APOC library is used to find the shortest path between nodes using Dijkstra's algorithm.

The node approach was the most appropriate since the API only gives access to the location of players every minute or specific relevant events.

Neo4j required very few changes, however, the process of rebuilding the old nodes and relations was the more time consuming of the necessary technologies adaptations due to versions compatibility of the graph database.

4.4.4 Materialize

Materialize (*Materialize* 2019) is a Cascading Style Sheets (CSS) framework to speed up the design process of the application offering a simple way to create the front-end frameworks, based on Material Design to reveal a simple aesthetic and efficient layout so the user can take full advantages of the capability without unnecessary distractions.

The interface of VisuaLeague III was completely redesigned, but materialize was kept as the main library to create intuitive layouts that would better apply to the new needs. Section 4.7 describes the choices in detail showing the complete interface.

4.4.5 Leaflet, noUISlider and ChartJS

Leaflet (*Leaflet* 2019) is an open source JS library for mobile friendly interactive maps which was used with the help of two plugins, MovingMarkers and Minimap. This was the choice due to its support onto non-geographical maps and imbued features like zoom interaction to have control of the size of the visualization.

NoUISlider (*noUiSlider* 2019) is a lightweight JS range slider library with useful features like different slider orientations, value ranges. Its highly customizable and takes major relevance when enabling user interaction with time or additional parameters, like setting the value for DBSCAN clustering algorithm parameters.

ChartJS (*ChartJs* 2019) is a library for charts in JS. Being very versatile in charts and giving options to mix and customize them. It was the best choice to attend user expectations when consuming static data. Some of the charts used from this library are bar charts, pie charts, radar charts, line charts and combinations between them.

Besides keeping their old functions in the animated map chartJS, leaflet and noUiSlider were also used to create graphs, interactive maps and sliders, respectively, in the new visualizations for each analysis method. Other options were considered, like D3.js (*D3* 2020), but ChartJS proved to do the necessary to create visualizations for game statistics while leaflet covered its faults when it came to representations in space over time. Section 4.7 specifies different applications of these libraries.

4.4.6 jDBSCAN

jDBSCAN (*jDBSCAN* 2020) is a repository that offers the DBSCAN algorithm in JS. The algorithm works with spatial data and also on data points given by GPS coordinates. It can be used on data with a time dimension also, allowing spatiotemporal clustering.

jDBSCAN was used, in VisuaLeague III, to create the option to cluster positions of players and events, in the multiple match variant of each analysis type. In the implementation, to the MinPts parameter was given a default value of two, making groups of only one position not possible. The Eps parameter is determined by user interaction, as explained in Section 4.7.6.

4.5 Data gathering

Riot API (Section 4.5.1) and DataDragon (Section 4.5.2) are official data repositories from Riot games. This Section presents a brief explanation of these repositories to understand future methodologies and limitations in VisuaLeague III. Pandascore (Section 4.5.3) is a third party API used to retrieve data about professional teams in League of Legends' eSports scope. This is is not directly related to Riot but was necessary since the official API does not give information about professional players nor teams. All of these sources provide data in JSON format. Section 4.5.4 describes and justifies the approach to store static data locally.

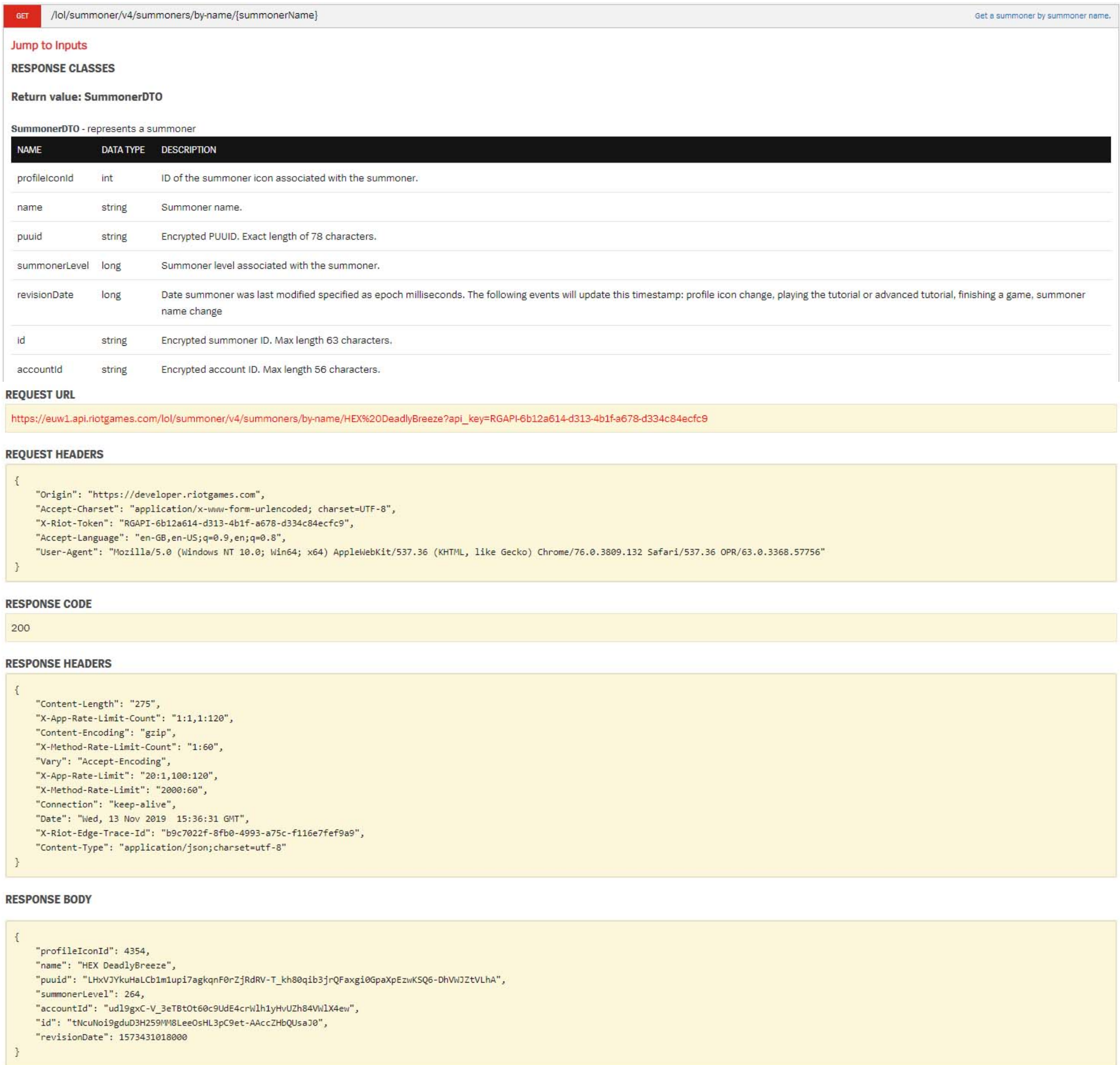
4.5.1 Riot API

Riot API is the main data source of VisuaLeague with a wide variety of information including player statistics, ranks, tournament results, match statistics and team statistics for different type of game-modes. To extract data from Riot is required an account with a request token that will give permissions to do queries. This token lasts for twenty four hours and currently it must be manually changed in the source code. Each user has two limited amounts of requests: one hundred requests every two minutes and twenty requests each second which can become restrictive since multiple requests are required to extract all required information of a single match. The objective of analysing aggregated matches will then become more challenging as the queries scale when searching for multiple games from multiple players.

Riot API separates data in different requests: requests with a player summoner name to retrieve the account id; requests for rank information for that account; requests for a match list, with a specific number of games, from that account; requests for statistical information of those matches; requests for spatiotemporal data of those matches.

Since VisuaLeague offers a player search based on summoner name and region, the main queries utilized in the project follow the approach of identifying an account id from those inputs. With the account id is possible to extract a match list which contains a defined number of the player's most recent matches, in a very summarized way. The quantity of matches loaded depends on the analysis type and filters from users, being possible to get up to one hundred matches in a single match list request. Through the game id acquired from a match, it is possible to request match data, where general statistics of that game are described like total gold for each team, or total kills for a player. In a separate request called timeline data is possible to get spatiotemporal data with information specific to players and events positions in certain time frames. Some incoherences in requests can be unintuitive to work with, for instance, to request match list data is necessary an encrypted account id and to request ranked data is necessary an encrypted summoner id.

Figure 20 shows Riot's API user interface to retrieve information of a specific summoner name together with its query response where the account id will be utilized for future requests. The three main datasets are further explained with the data process in Section 4.6 together with examples of the resulting files after being processed by VisuaLeague III.



GET /lol/summoner/v4/summoners/by-name/{summonerName} Get a summoner by summoner name.

Jump to Inputs

RESPONSE CLASSES

Return value: SummonerDTO

SummonerDTO - represents a summoner

NAME	DATA TYPE	DESCRIPTION
profileIconId	int	ID of the summoner icon associated with the summoner.
name	string	Summoner name.
puuid	string	Encrypted PUUID. Exact length of 78 characters.
summonerLevel	long	Summoner level associated with the summoner.
revisionDate	long	Date summoner was last modified specified as epoch milliseconds. The following events will update this timestamp: profile icon change, playing the tutorial or advanced tutorial, finishing a game, summoner name change
id	string	Encrypted summoner ID. Max length 63 characters.
accountId	string	Encrypted account ID. Max length 56 characters.

REQUEST URL

https://euw1.api.riotgames.com/lol/summoner/v4/summoners/by-name/HEX%20DeadlyBreeze?api_key=RGAPI-6b12a614-d313-4b1f-a678-d334c84ecfc9

REQUEST HEADERS


```
{
    "Origin": "https://developer.riotgames.com",
    "Accept-Charset": "application/x-www-form-urlencoded; charset=UTF-8",
    "X-Riot-Token": "RGAPI-6b12a614-d313-4b1f-a678-d334c84ecfc9",
    "Accept-Language": "en-GB,en-US;q=0.9,en;q=0.8",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.132 Safari/537.36 OPR/63.0.3368.57756"
}
```


RESPONSE CODE

200

RESPONSE HEADERS


```
{
    "Content-Length": "275",
    "X-App-Rate-Limit-Count": "1:1,1:120",
    "Content-Encoding": "gzip",
    "X-Method-Rate-Limit-Count": "1:60",
    "Vary": "Accept-Encoding",
    "X-App-Rate-Limit": "20:1,100:120",
    "X-Method-Rate-Limit": "2000:60",
    "Connection": "keep-alive",
    "Date": "Wed, 13 Nov 2019  15:36:31 GMT",
    "X-Riot-Edge-Trace-Id": "b9c7022f-8fb0-4993-a75c-f116e7fef9a9",
    "Content-Type": "application/json;charset=utf-8"
}
```


RESPONSE BODY


```
{
    "profileIconId": 4354,
    "name": "HEX DeadlyBreeze",
    "puuid": "LHxVJYkuHaLCb1m1upi7agkqnF0rZjRdRV-T_kh80qib3jrQFaxgi0GpaXpEzwKSQ6-DhVWJZtVLhA",
    "summonerLevel": 264,
    "accountId": "udl9gxC-V_3eTBtOt60c9UdE4crWlh1yHvUZh84VWlX4ew",
    "id": "tNcuNoi9gduD3H259MM8LeeOsHL3pC9et-AAccZHbQUsaJ0",
    "revisionDate": 1573431018000
}
```


Figure 20: Riot's API example of a response to a request. (Chikish, Carreras, and Garci, 2019)

4.5.2 DataDragon

DataDragon (*DataDragon* 2019) is an official web service provided by Riot games apart from Riot API to make static data available handling every information unlikely to change often, in order to reduce usage and maintain server stability. Usually this data is updated when a new game patch arrives, but it doesn't always keep the pace with the current build of the game making some slight inaccuracies in analysis after recent updates. It offers two ways of access the data, one can opt to download the files or access them directly through

Uniform Resource Locator (URL). In VisuaLeague III both approaches are utilized. Big datasets like champion information, map ids, items, masteries and summoner skills are saved locally (Section 4.5.4). When it is necessary to access specific data like information to a specific champion skill of a particular match, or a champion's image, a direct request through URL is used instead.

LoL has patches with major changes every season (every year) and small updates between them. When VisuaLeague III started to be built, it was necessary to make a lot of adjustments in the previous version to assure that every information was being processed as intended because a lot of the structure presented in JSON documents from static data repositories changed, including the way to access champion and runes ID which contributed to a major debugging phase in the project. Besides that was often unintuitive to find data due to small incoherences, for instance, some champion names in the databases are different from names in game. Other example is that to request data relative to champions, in some requests (like portrait images, for example) is necessary to use the champion name, but champions in data from matches are referred by a numeric id. This makes conversions between champions' ids and names necessary throughout the application.

4.5.3 Pandascore

As mention in Section 3.6, one of the project's objectives is to support multiple games for professional teams. Pandascore is a database that stores data about the competitive environment of several games relevant to eSports including, Counter-Strike, PubG, DOTA 2, Overwatch and League of Legends.

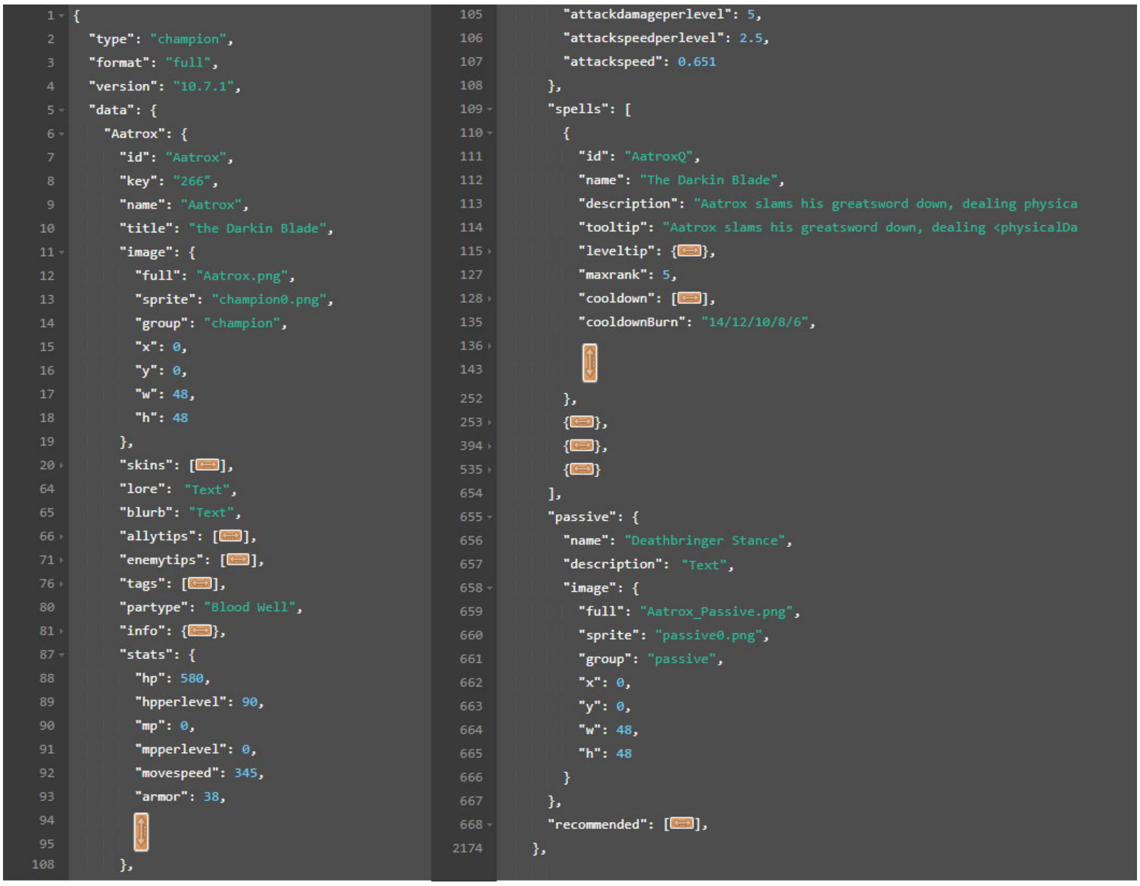
Just like Riot API it requires a token associated to a user account. In this case, the limit is one hundred request per hour for free users, which does not affect VisuaLeague III as it needs only a single request to get all data about professional teams and their respective players being then stored in couchDB and updated only when necessary.

There are other databases options with APIs that have information relative to LoL' professional matches and eSports, for instance Esport API (*Esport API* 2019). However, Pandascore was nearly the only one that offered a free option to utilize the data that could fulfill our purposes with information about professional teams and players. Some of these paid subscription databases also provide data about matches of professional gameplay including eSports tournaments, sometimes including real time tournaments' spatiotemporal data, like GameScorekeeper (*Gamescorekeeper* 2019).

4.5.4 Local files

Local files store static data in JSON format, so it does not need to be constantly loaded from the web. This makes VisuaLeague III more optimized and also reduces the amount of requests to Riot's servers. Those files are then directly used by the application and can be manually updated to newer versions through server routes.

To keep track of compatibility there is a main file containing the most recent version of the data which is used to request the remaining files, maintaining coherence and giving the option to be compared with the DataDragon version. Figure 21 shows an example of a file storing champions' information then used to create associations of those ids to their names and images throughout the application. In order to save space the content was slightly condensed and some values edited in the figure as it is meant to give an overview of the file structure. Besides the champions there are files for runes, items, summoner spells, maps and constants, the last one standing for the ids of different game modes, which follow the same JSON structure once stored in couchDB.




```
{
  "type": "champion",
  "format": "full",
  "version": "10.7.1",
  "data": {
    "Aatrox": {
      "id": "Aatrox",
      "key": "266",
      "name": "Aatrox",
      "title": "the Darkin Blade",
      "image": {
        "full": "Aatrox.png",
        "sprite": "champion0.png",
        "group": "champion",
        "x": 0,
        "y": 0,
        "w": 48,
        "h": 48
      },
      "skins": [],
      "lore": "Text",
      "blurb": "Text",
      "allytips": [],
      "enemytips": [],
      "tags": [],
      "partype": "Blood Well",
      "info": {},
      "stats": {
        "hp": 580,
        "hpperlevel": 90,
        "mp": 0,
        "mpperlevel": 0,
        "movespeed": 345,
        "armor": 38,

      },
        "attackdamageperlevel": 5,
        "attackspeedperlevel": 2.5,
        "attackspeed": 0.651
      },
      "spells": [
        {
          "id": "AatroxQ",
          "name": "The Darkin Blade",
          "description": "Aatrox slams his greatsword down, dealing physica
          "tooltip": "Aatrox slams his greatsword down, dealing <physicalDa
          "leveltip": {},
          "maxrank": 5,
          "cooldown": [],
          "cooldownBurn": "14/12/10/8/6",

        },
        {},
        {},
        {}
      ],
      "passive": {
        "name": "Deathbringer Stance",
        "description": "Text",
        "image": {
          "full": "Aatrox_Passive.png",
          "sprite": "passive0.png",
          "group": "passive",
          "x": 0,
          "y": 0,
          "w": 48,
          "h": 48
        }
      },
      "recommended": [],
    },
```


Figure 21: Example of a file containing champion data.

4.6 Data processing

This Section will discuss the process of preparing the data utilized in the different types of analysis in VisuaLeague III, focusing in what happens on the server side of the web application. Starting with Section 4.6.1 where is shown the way to handle data for single player analysis; Section 4.6.2 describes how the players from teams are found and respective matches loaded; Section 4.6.3 talks about the differences from custom games' matches data relative to other game modes; Section 4.6.4 describes the preprocess for multiple games for one or more players.

When requesting data to Riot API by summoner name and region there are three major datasets of interest:

Match list — Dataset containing recent matches of a searched summoner. The request can be sent without any additional attributes where a list of the most recent one hundred games will be loaded. It is also possible to filter requests like amount of games and game type which VisuaLeague III uses in aggregations, Section 4.6.4. The main attributes of interest in this dataset are ”gameId” to identify the game in future requests, ”champion” to know which champion the player used in case filters need to be applied, and ”timestamp” to chronologically order games, shown in Figure 22 at the left.

Match data — Dataset where is gathered match information from a given game id. General statistics of the game can be found here with representations of totals achieved from each player like ”total damage dealt to champions”, ”total gold collected” and ”total minions killed”. The ”gameDuration”, ”items”, ”runes” and ”version” are important attributes as they contain the information to build a context in the analysis, shown in Figure 22 at the right.

Timeline data — The major source of spatiotemporal data has an example in Figure 23, used to build all visualizations with spatial relations like the animated map, for single match analysis, and maps & timelines, for aggregated analysis. This dataset offers positions of players in sixty seconds intervals and registers of events like kills, elite monsters kills and turrets destruction.

To get all data required for VisuaLeague III, usually, all these datasets must be requested, with the exception of custom games where match list data is not required since the game ids are previously given by the user (explained in Section 4.7.5).

4.6.1 Single player data

Data related to a single player is the most straightforward since it is Riot API main focus. With user inputs of summoner name and region VisuaLeague III will check in couchDB if there is available data for that player. In case there is data stored with less than thirty minutes old it will be directly loaded from couchDB. If there is no data for that player and region available, or the data is older that thirty minutes, a new request to Riot API is made adding the information and replacing the old one, if existent. Thirty minutes is the defined time for a match data to become obsolete, since it is an average match length it is unlikely to have new data before and very likely to have new data after that time if the player is making matches.

To get new data, VisuaLeague III summoner model will take the response from the previous mentioned request and extract general information about player's account, pass-

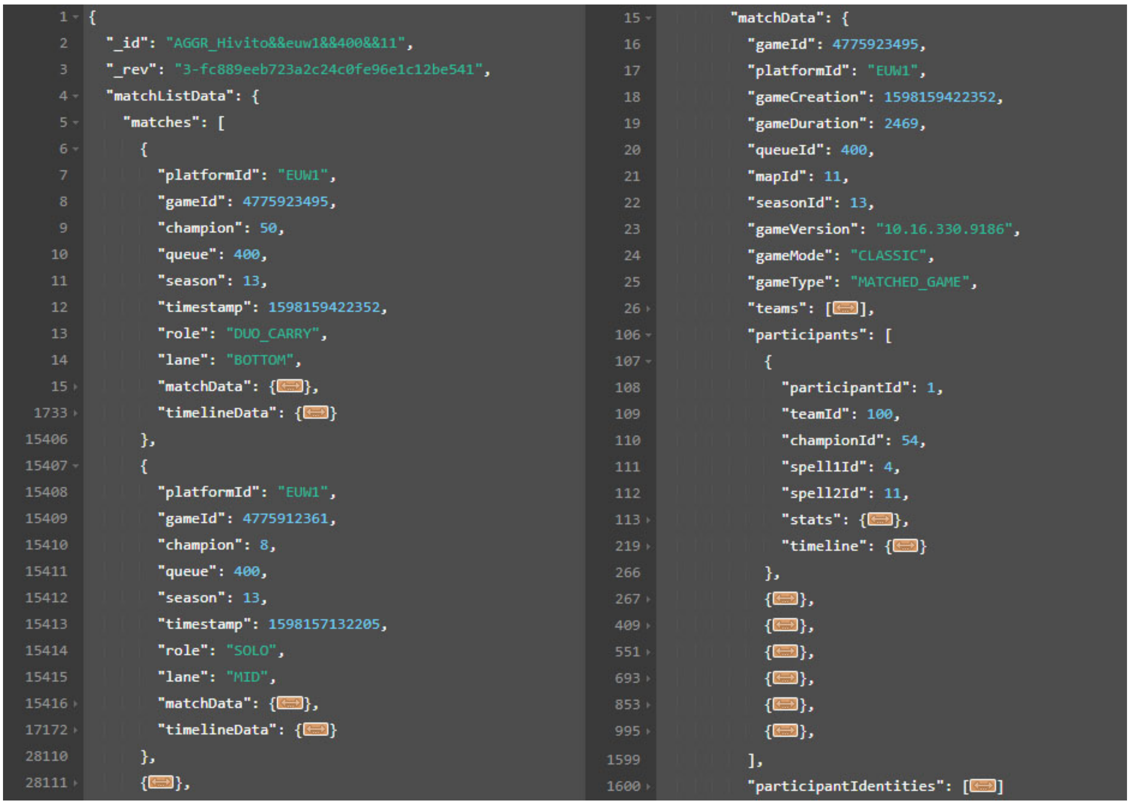


Figure 22: Example of a JSON file storing data retrieved from Rito API. Left: match list dataset. Right: match data dataset.

Figure 23: JSON file example of timeline dataset.

ing the received account id on to the match model. Here match list data is requested and the ids of each match received are used for match and timeline data requests, separately. After getting all the data, an object in JSON format is sent to couchDB where is stored and passed to single player's front page creating the visualizations described in Section 4.7.5.

4.6.2 Teams data

Riot API does not support team searches because teams are extremely mutable and players usually do not train as a team. Normally players expand their abilities in solo queue matches which are individual games with a rank system to determine player quality. To simulate a team search, players are defined by the user individually (detailed in Section 4.7.4). VisuaLeague III then requests match list data for all selected players in one defined team using their summoner ids and regions. Gathered match ids are compared between players, keeping only ids present in every player match list to assure that unnecessary data is not loaded. The process after selecting match ids is similar to the one described for the single player data, with the exception that in couchDB a unique id is kept to reference the team name and the selected members, since the members in a team can change.

4.6.3 Custom games data

Custom games are a special case as there is a lot of limits in information access. First, it is impossible to find a custom game through match history of a player, which means that custom games' data will not appear when making requests to Riot API by a match list request. To overcome this problem, together with the team coach, it was created a solution by using a specific match URL to search for a match directly. Only players who participated in that custom game have access to the match URL, but this is not a problem for coaches and professional players, since they usually keep those from scrims. To get data of custom matches, VisuaLeague III has a dedicated interface where the user inserts the id or URL from a custom game. In the second case, the URL is automatically transformed by the application to find custom match game's id, requesting match and timeline data to Riot API and proceeding as a single player process (Section 4.6.1).

However, there are still some limitations in the data. The major one is an absence of information about players' accounts who participated in the match. For that reason, it is impossible to identify a player id and to relate a champion from a match. This makes some aggregations, filters and visualizations not viable. Figure 24 at the right shows a data fragment of player information in a solo queue ranked game, versus a custom game at the left, for reference. In single game analysis this absence is not a problem, since the professional coach stated that although it would be useful to identify players, usually, when analysing that type of games, coaches and players already know who participated and can easily relate the players to their champions and roles in the team. In aggregated games visualizations however, this requires additional inputs from the user about team side (red or blue team) and win state (victory or defeat). To help VisuaLeague III determine which of the teams present in match data refers to the user team. Only links from one of the sides and one of the win states can be analysed at once.


```
"participantIdentities": [
    {
        "participantId": 1
    },
    {
        "participantId": 2,
    },
```

```
"participantIdentities": [
    {
        "participantId": 1,
        "player": {
            "platformId": "EUW1",
            "accountId": "xMh53cZxwYVhaKy2rBZp5cRny0FaN8KfbjUiSA-4QkQdkg",
            "summonerName": "1 Faker ",
            "summonerId": "XUlDGlqP9JmIQYxlBV6Fli0DRTseKiwD7cc-Qy0nyz842TM",
            "currentPlatformId": "EUW1",
            "currentAccountId": "xMh53cZxwYVhaKy2rBZp5cRny0FaN8KfbjUiSA-4QkQdkg",
            "matchHistoryUri": "/v1/stats/player_history/EUW1/37820897",
            "profileIcon": 4270
        }
    },
    {
        "participantId": 2,
        "player": {
            "platformId": "EUW1",
            "accountId": "fzrIc1N8ZQIfCY-ANueSy3Xp7UJyfdga091sdWPK-B2WEp2fIF1w2oK5",
            "summonerName": "YooLek",
            "summonerId": "aohZqpM7V0nVWykeEA0n1D2me3FCVfaCiHwPuw5G0S4Vjssg",
            "currentPlatformId": "EUW1",
            "currentAccountId": "fzrIc1N8ZQIfCY-ANueSy3Xp7UJyfdga091sdWPK-B2WEp2fIF1w2oK5",
            "matchHistoryUri": "/v1/stats/player_history/EUW1/2267391702462560",
            "profileIcon": 4490
        }
    },
```


Figure 24: Comparison of player account identities. Left: Custom games Right: other game types. Image extracted directly from Riot API (Riot Games, 2019b) and edited.

4.6.4 Multiple matches data

To get data for multiple matches, filters are applied to searches via user input, to retrieve the games one's interested. One filter acts on the amount of games requested, as the amount can vary according to analysis purposes. There's a filter to select the game mode (usually five versus five in Summoner's Rift), since it does not make sense to analyse aggregated games from different game modes due to their disparity in strategies. The same applies to match types, where the user can select one of the available options, like ranked games. This type of filters will be necessary in single player analysis and team analysis.

The number of games to load for each analysis varies from two to one hundred games. The max amount comes due to limits in the request for a match list. It would be possible to expand the max number of games over one hundred but it would increase the waiting time for the requests to complete. Additionally, the professional coach stated that it is very unusual for a player, or coach, to analyse that many games anyway being twenty to forty a reasonable number since it is the average of games they usually have in consideration for solo queue.

This filters will then be passed in the request string to Riot API getting match list data. As Riot API separates the data for a match in more than one request, it is necessary to get all the different parts of the data, namely the match data and timeline data, fusing them together in a single object for each match. Some specific information can also spend more requests per match, for example data relative to the players ranked statistics would mean an addition of ten requests for each match, since it would require a separated request for each player in each game. For this reason this feature was not implemented in the current version of the application. Once the data is assembled, all matches will be stored in couchDB with a unique id representing filters selected by the user and then loaded this time, and every time the same filters are applied, until the time limit expires.

For multiple matches in team analysis it is necessary to determine which summoners are requested by the user in the team selection, as described in Section 4.6.2 and then use the same process for player analysis. When stored in couchDB the id will have references to the name of players selected since they can vary inside a team.

In the case of multiple matches in custom analysis, the user is required to have multiple URLs from custom matches and requests will be done to get data directly from the games' ids extracted from those URLs.

4.7 Design

This Section describes VisuaLeague III prototype design where all the major design decisions will be discussed and justified. All the design and user experience decisions were based on interactive informal interviews with a professional coach, helping to fulfil the requirements of professional LoL analysts.

First in Section 4.7.1 an overall explanation of design choices in the application will be discussed together with motives to change from the previous implementation. The animated map visualization will be explained in Section 4.7.2 as it is used in three of the analysis methods. VisuaLeague III supports three types of analysis: Player analysis (Section 4.7.3); Team analysis (Section 4.7.4); Custom analysis (Section 4.7.5). Finally, the approach to select multiple matches for each of the previous methods is discussed in Section 4.7.6 as it behaves similarly to the three different options with small adaptations.

4.7.1 Overall design

The previous prototype of VisuaLeague was developed only considering analysis for a single player and a single match with the animated map as main visualization therefore in order to include multiple game analysis and team selection, it was necessary a restructure of the layout. Figure 25 shows the new layout condensed in the top of the browser page where a navigation bar takes all the user interface leaving space to show visualizations in the center of the screen. New choices between different analysis methods were added to better suit user's needs for different types of games.

The main categories are player analysis, team analysis and custom analysis. In the first, the user searches for a player only loading recent matches for that player discussed in Section 4.7.3. In team analysis, the user can create a team and search a selected group up to five players, or rely on Pandascore's database to choose a predefined professional team to look for professional players available, detailed in Section 4.7.4. Finally a custom analysis menu which is exclusively dedicated to custom games due to their differences in data availability compared to the rest of the game modes, detailed in Section 4.7.5. Custom games analysis is where scrims of professional players can be accessed as they are considered a custom game by Riot.

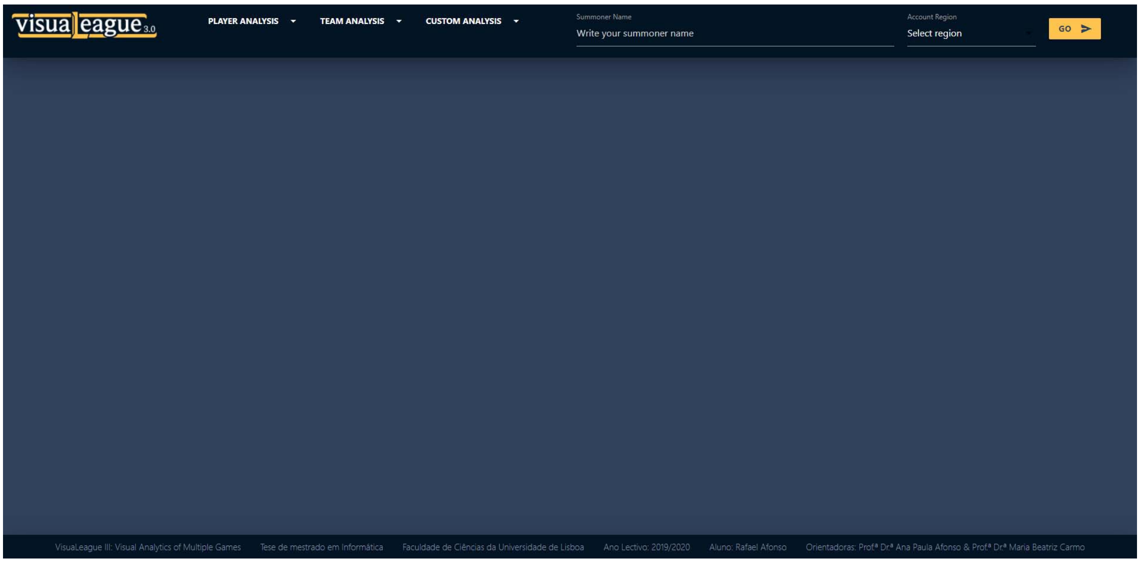


Figure 25: VisuaLeague III front-page without any loaded data.

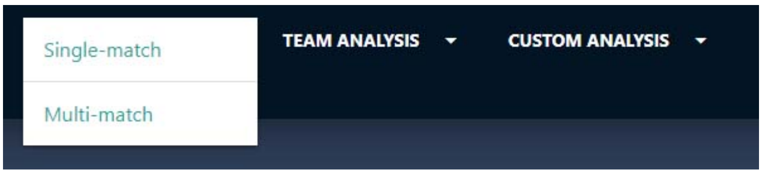


Figure 26: VisuaLeague III menu for single and multi-match analysis.

When selecting one of the three choices above, the user will opt by a single-match or a multi-match analysis as shown in Figure 26. The first option, presents visualizations with several game statistics for the selected player, treating each match individually to enable single match analysis. The user can further analyse relevant games with the animated map for more detailed visualizations, if the general information presented in overviews is not sufficient. The multi-match analysis, based on user defined filters (like number of games and game type), provides visualisations of statistical and spatiotemporal information to enable analyses of players and teams over multiple games.

The visualization techniques will be displayed in the blue area bellow to the navigation bar (Figure 25). The blue color scheme was kept from the previous prototype as it is coherent with game colors. However, the salmon color used in the buttons, as well as some hover colors, were changed to better fit League of Legends' color scheme, as it is mainly focused on deep blues and golden yellows. A Logo for VisuaLeague was developed to give character considering the rest of the layout colors and bring balance to the yellows used in the buttons.

4.7.2 Animated map

The animated map, being the only visualization technique in VisuaLeague III that opens in a separated view, is a real time representation of a chosen game. This type of visualisation is effective when the objective is to analyse a specific occurrence in a match allowing to re-watch positions of every champion in the game along with respective scores, statistics and gold. Figure 27 shows the overall layout of the visualization where the map from LoL is used to show Nexus, inhibitors, towers and positions of the champions selected by players using their respective image profile. Each of the displayed elements are updated according to occurrences in the game being possible to observe players constantly moving and events happening at their respective time stamps. This feature was barely changed in terms of design and approach, but it was necessary to update all data and create new methods to take into account changes in DataDragon's JSON files and in Riot API according to League of Legend's new season.

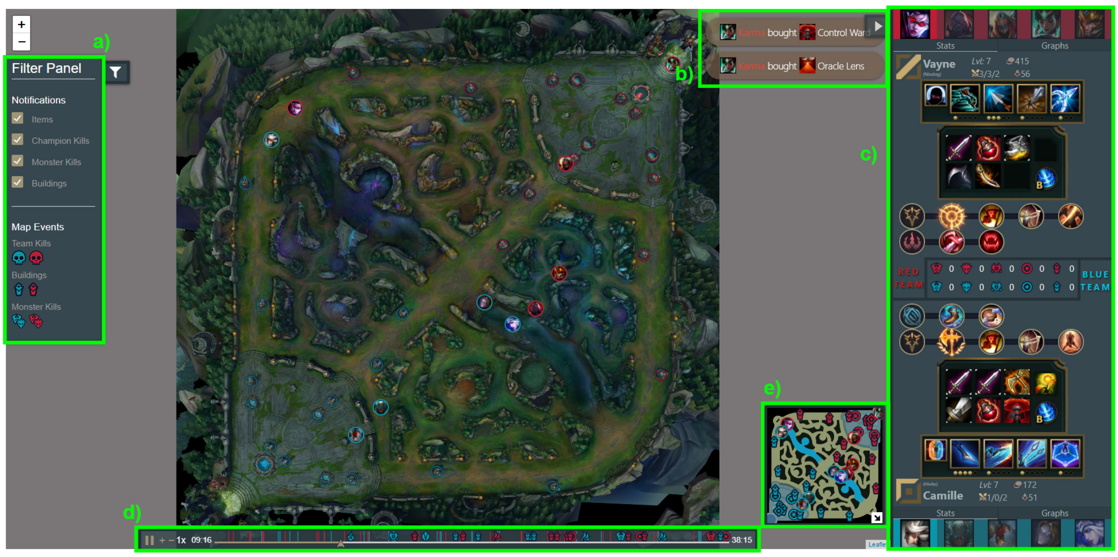


Figure 27: VisuaLeague III animated map visualization. a) filter for events displayed in timeline. b) event notifications. c) statistics of game players and teams. d) timeline with events. e) minimap.

Figure 27 e) shows a minimap which serves for the user to keep track of the general match composition in case a high level of zoom is used in the main map. The same Figure 27 a) shows a filter menu where events can be opted to hide or show, being all shown by default. If the analysis tends to focus on the blue team it is possible to hide all the representations relative to the red team and see only what is intended. Same happens with more specific filters like kills, monsters and towers.

Bellow the map (Figure 27 d)) is the timeline where the user can speed up or slow down the time in which the game displays events and positions and is capable of going back and forth in game time avoiding watch the whole match. It is also possible to pause the time for a deeper analysis of that time frame. Major events like tower destruction and elite monsters are displayed in the timeline to facilitate the identification of important objectives.

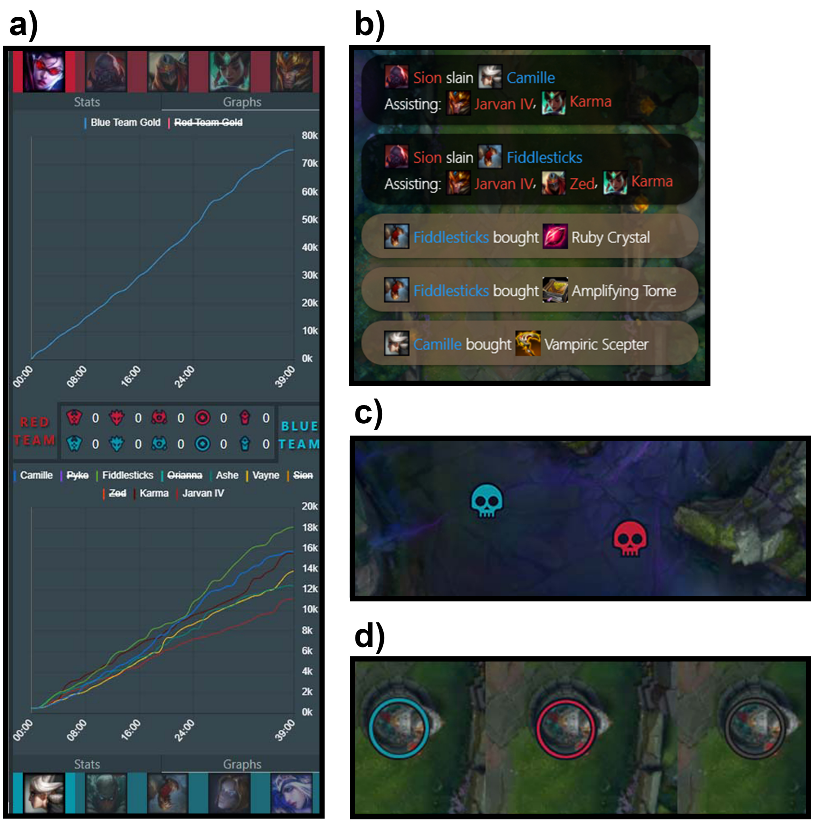


Figure 28: Animated map visualization details. a) statistics of gold for team and player comparison. b) different events notification for kill events and item bought. c) kill events displayed on the map. d) displaying tower state, color circles represent each team and become grey when destroyed

Figure 27 c) shows multiple representations of important factors to have in consideration in a match. The main one being an overall state of player's build which is the major reference to know how someone is performing by reflecting upon in-game time and items present or KDA scores from him or the team. The user can freely choose players to compare, making comparisons from two champions of the same role easier to access. Figure 28 a) shows more options available on this tab with statistics showing gold for team and each player's champions represented by graphs using ChartJS library, Section 4.4.5, where the analyst can more efficiently compare teams and players farming efficiency.

To better keep track of events and help the user realise something important happened, small popups will appear at the top right corner of the map area distinguishing different occurrence type by colour as shown in Figure 28 b). Also kill events will be marked on the map for a period of time with the respective team colour, Figure 28 c) and towers destroyed will show grey circles around them instead of their team color version, Figure 28 d).

4.7.3 Player analysis — single-match

The Player single match analysis focuses on individual matches for one player where the user is able to track games of a specific person by introducing the summoner name and choose one of the available regions in the search bar.

VisuaLeague III displays the most recent matches of the player with general information about each game giving the user an overall idea of player performance, Figure 29. A new layout was developed to display player and matches information, as the professional coach stated that some important information was missing. Namely the rank of the searched player together with LP was very important for him to keep track of player's level in the ranked solo queue.

The interface of Figure 29 is the result of this change, being divided in two sections, a smaller one on the left presents information related to the selected summoner account with account name, level and rank for the current Leagueand a second section on the right show a list of games. This list displays information about the champion profile image, obtained level inside the game, game win state, player assigned position, KDA, map type, date, gold and farm. This information is important to determine if a certain game is relevant for further analysis. It is then possible to click in the rightmost yellow button to further analyse a match with the animated map referred in Section 4.7.2.

4.7.4 Team analysis — single-match

Team analysis single match offers two different options: user customized teams (Figure 30 top) and pandascore pro teams (Figure 30 bottom left). The first is meant for the user to build teams where the s/he has full control about their players. This gives the ability to be more flexible and coaches/analysts can include players who are not in the professional environment yet or professional team of less popular Leagues. The process of building a team will start when clicking the button to add a team at the top right. Figure 30 at the bottom right shows the interface where the user will input a team name

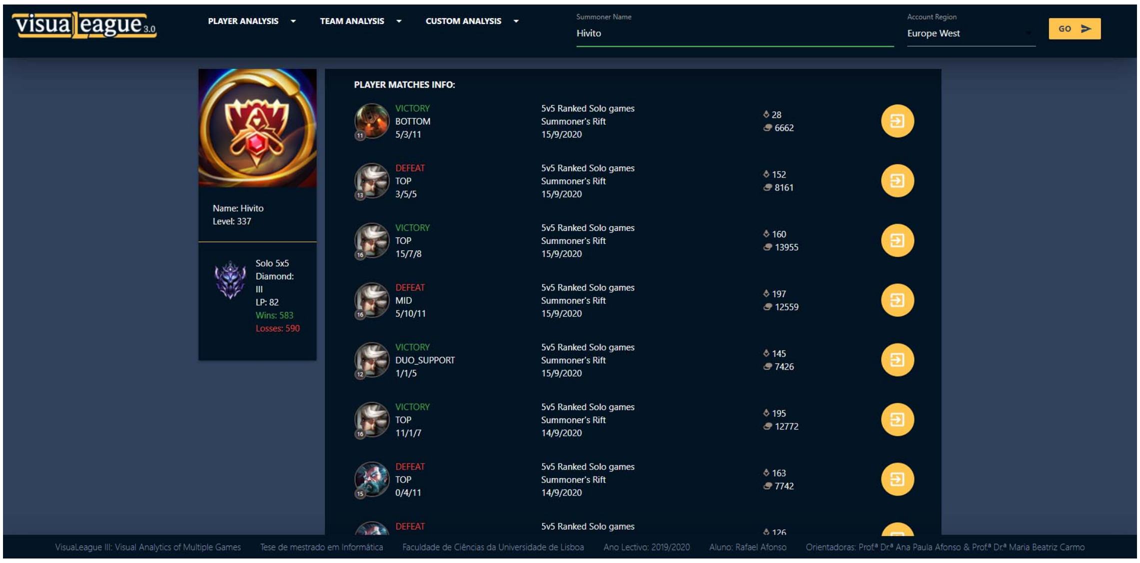


Figure 29: Player analysis for single matches. Display of player recent games.

and include players. To add a player it is necessary to write the summoner name and respective region. Then the yellow icon will check if a summoner exists as a player for that region. The button will turn green and the checked player will be added to team's members if existent, or the button will turn red and will not be considered in the team if that summoner name doesn't exist for that region. The number of summoners to build a team is arbitrary, as it can have five main players and many substitutes. Also smaller teams must be considered with less than five players. Once all summoners are verified, the user can finish the process by clicking create team and it will be uploaded to couchDB being kept and loaded for that page every time until it is deleted.

All the available teams will show in a collapsible visualization, displaying team name and number of members information when closed and every player available when expanded. Figure 30 at the top shows one opened team with current players icons, name and level. In the expanded view, the user has two options: choose the magnifier icon next to each summoner, searching for that player games directly; select an arbitrary number of players, from two to five, and search for matches for the group by clicking in the icon with a group and magnifier near the team's name. If the games loaded are for a single player, the layout is equal to the one presented for single player analysis but condensed in the right side of the page meaning that all summoner information will show in the header (Figure 30 top). However, if the games are for a team, a new layout is displayed as presented in Figure 31 with information focused on the whole team instead of a single player. The user can choose a game to get detailed information. Figure 31 on the right shows this visualization where every game is displayed individually with match statistics related to summoner spells, builds and runes, while also including scores like gold acquired and damage dealt. To analyse a match is possible to dive into the animated map visualization for a chosen game with all the players selected using the right most button in each match

(Section 4.7.2).

The second option in team analysis, pandascore pro teams, provides information from Pandascore repository (Section 4.5.3). In this option, an user can search for a specific professional team and have access to all its players. As explained before, information about professional teams is very mutable, not allowing team searches to be fully dependant on a repository, but it provides searches of specific professional teams or to make fast comparisons with professional players.

Figure 30 bottom left shows an example of a professional team directly extracted from Pandascore's API where team name, logo and team location is presented. If an user selects a team, the interface will expand and players will be shown just as in user customized team but with they personal profile pictures, lane and name. It is then possible to look for games for that player using the same magnifier icon or analyse the player profile page in Gamepedia's website, where coaches rely to get a more deep insight about professional players and teams. The layout for loaded games is similar to the one presented for user customized teams.

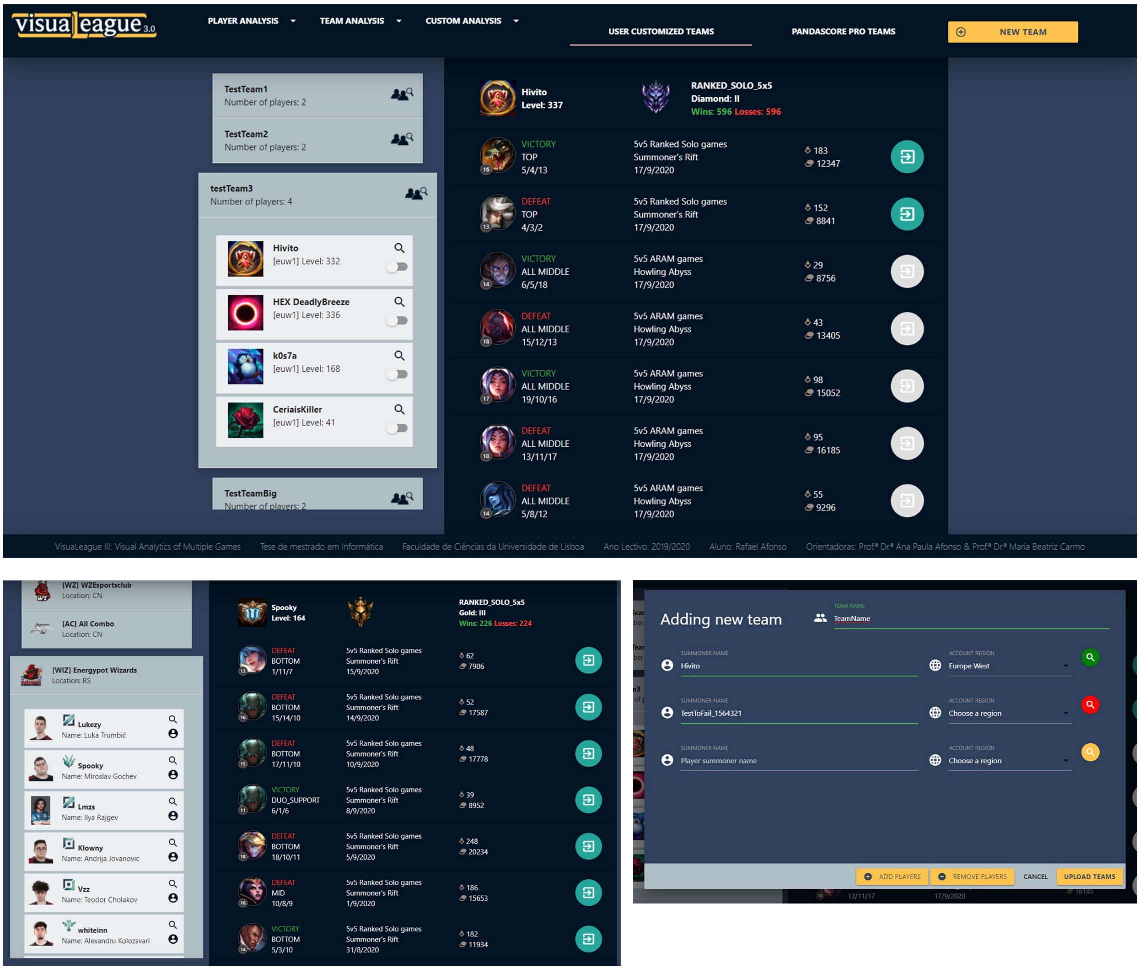


Figure 30: Team analysis. Top: User interface for customized teams. Bottom left: User interface for Pandascore's professional teams. Bottom right: User interface to add a team.

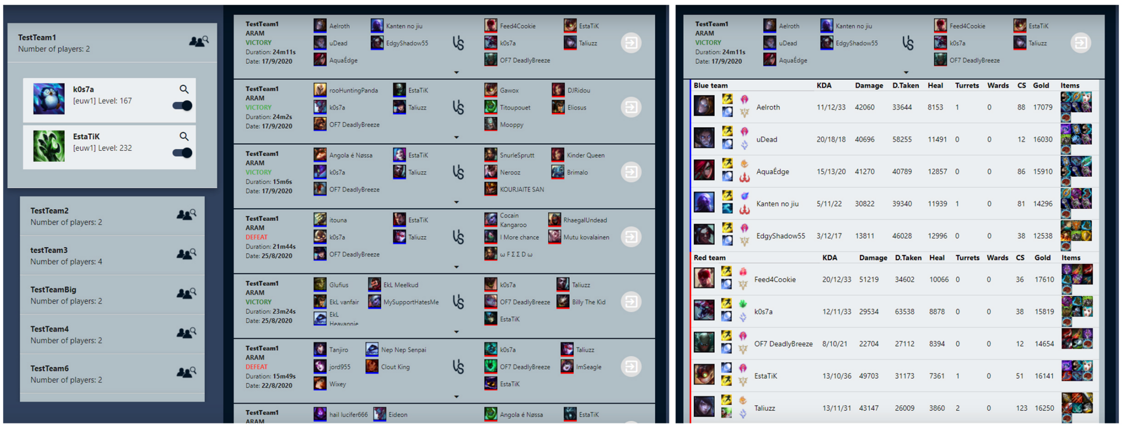


Figure 31: Team analysis visualization for each game loaded for selected players. Left: default state with general team information for each game. Right: detailed information of a selected game.

4.7.5 Custom analysis — single-match

In custom games analysis the user must have an URL (Figure 32). After searching for the match, VisuaLeague II presents detailed statistics for selected matches, unlike the previous iterations on single player and team games. The reason for this choice comes from custom games' data access restrictions and the current absence of tools that grant access to them. In the opinion of the professional coach, this makes VisuaLeague III a better tool for scouting. Also with custom games restrictions there is no way of knowing if the link is from the pretended match, therefore making this preview useful as it displays participants for both teams.

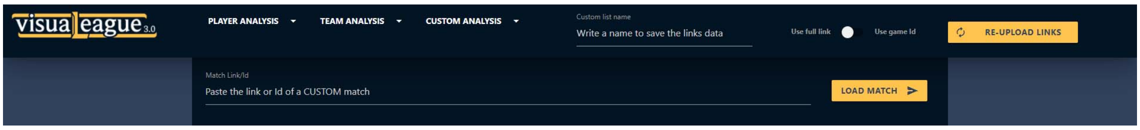


Figure 32: User interface to introduce a custom match link or id for analysis.

In Figure 33 at the top is a general information section about teams and bans of the selected match. Starting by giving context of the game with the game type, map, date and match duration. Teams' data is presented divided by colors (blue team and red team) where the user knows the winner and champions banned for each team. General statistics for each player build choices, like items, summoner spells and runes, are available as those are essential information to consider in analysis.

Bellow is a damage bar chart with damage counts from the game showing damage dealt, damage taken and healing done. Those are often used to determine player performance. The bar chart was the chosen graph as it is a visualization LoL's players and analysts are used to when it comes to damage meters allowing a fast perception on who has the highest value. It is possible to interact with the information, displaying one of the three available values or a combination of them. This allows coaches to adapt to different

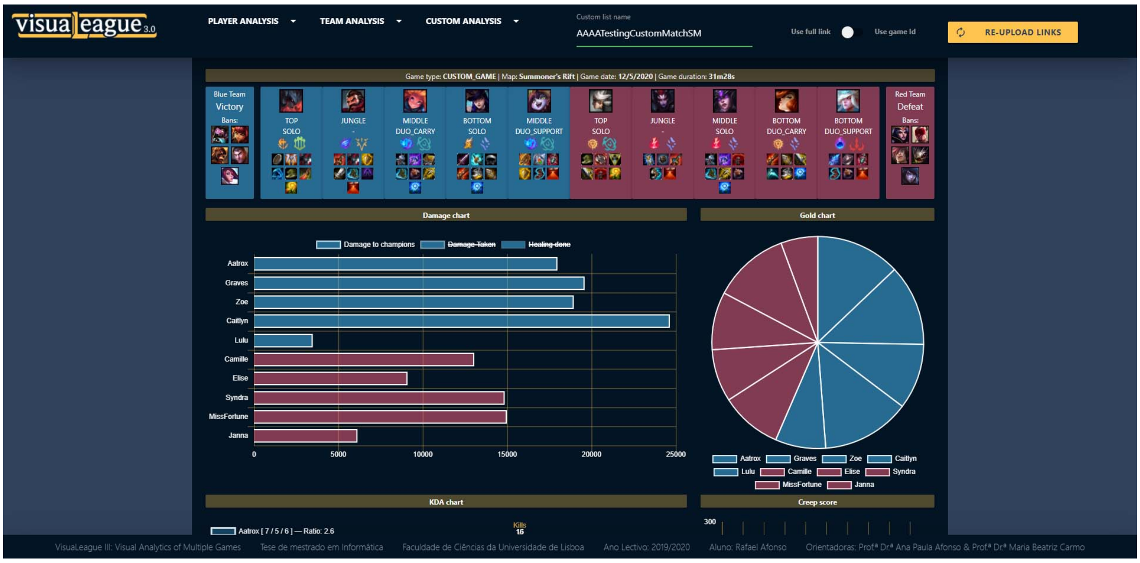


Figure 33: Information of a custom game.

player roles, for instance if the analysis focus is on the support, only healing done would be relevant. If the analysis has focus on a fighter tank role, damage done to champions and damage taken would be both relevant, thus analysed together.

Right next to it is the gold pie chart, to keep track of the amount of objectives a player is completing, since each objective or kill rewards players with gold and this is the main resource to become stronger in the game. A pie chart was chosen as it gives a better perception of team totals, making it possible to understand immediately which team got more gold just by watching which colour has a bigger portion of the pie. It is also possible to analyse individual values by hovering the mouse in each player's part in order to compare each champion. The user can interact with the pie by hiding or showing each player available. This allows to focus the analysis on specific players, for instance, select players of the same role and directly compare the amount of gold each one collected.

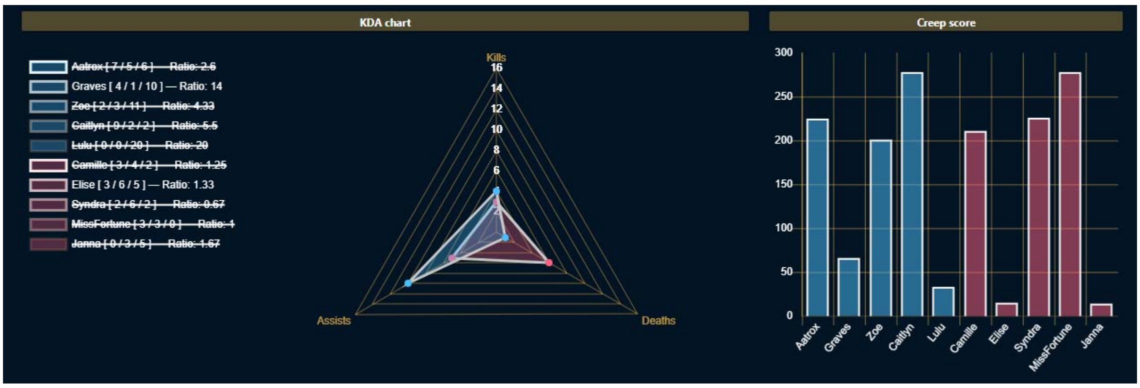


Figure 34: Custom analysis visualizations. Left: Graph of the player's kills, deaths and assists. Right: creep score.

In custom analysis it was decided to include a triangular radar chart (Figure 34 at the left) to provide a visualization that relates the three KDA values. This grants the analyst the ability to easily compare individual values by reading one of the axis. It is also possible to analyze overall performance by comparing the colored areas, where the perfect performance would make the colour grow only between kills and assist, leaving deaths empty. The information is displayed for each participant and can be interacted with, giving the ability to directly compare players from the same roles. In the example (Figure 34 at the left), is a comparison between the two junglers of that match. It is immediately recognizable that the blue jungler had better performance as the blue colour only grows to the kills and assists side of the chart, while the red player grows to the deaths attribute.

The creep score chart shows the number of minions killed by each player on that game. Depending on how long the game is, an experienced player, analyst, or coach, can immediately tell if a summoner did well in that game, just by looking at the creep score for a specific role. The bar chart was once again chosen as the main purpose is to observe individual values.

Finally the user can choose to access the animated map visualization to analyse spatiotemporal data for the selected match. The animated map behaviour is identical to what was explained in Section 4.7.2 except that the names of the players will not be present due to the custom game data limitation referred in Section 4.6.3.

4.7.6 Multiple match analysis

In multiple match analysis for a single player, initially, the user searches for a summoner and region and a new view (Figure 35) is presented with some filters to customize the search: number of games, type of game, and champion. The number of games is limited to one hundred games (Section 4.6.4). The type of game is also required since, in the case of creating aggregations for multiple games, it would not make sense to group matches from different game modes. The user can also filter matches to load only a certain champion to get player performance every time s/he plays that character.

Multiple match analysis for custom matches is an exception. Since custom match data is not available in match list data, the user must input URLs for every game s/he wants to analyse. For the fact that player identity is also not available for custom matches, the user must previously separate matches according to team side and game result and indicate in the interface that information (Figure 36).

Once the data is loaded, VisuaLeague III shows an interface composed by three main tabs (Figure 37): Selected Player, All Players, and Maps & Timelines. The first, displays visualizations with information related with the selected player or team. The second tab, All Players, displays visualizations considering all players or teams present in the matches. The last tab, Maps and Timelines, displays spatiotemporal data using maps where the user can interact with filters to customize analysis.

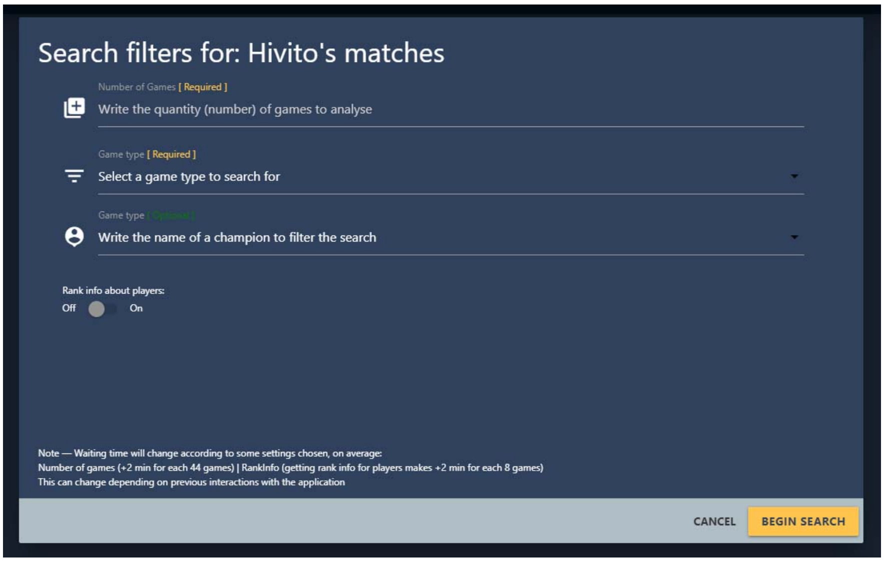


Figure 35: Search filters for aggregated matches in player and team analysis.

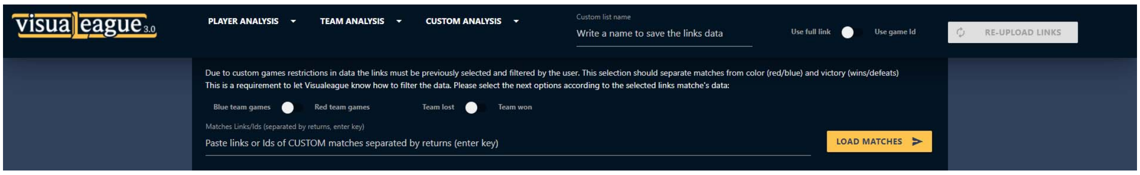


Figure 36: Search filters for aggregated matches in custom analysis.

There are two side panels that can be opened or closed. The left panel gives additional options do interact with the visualizations and the right panel shows additional information for each visualization describing what it is and how it can be interacted with. Both panels change according to the current tab being displayed.

Selected player tab

The Selected player tab offers different visualizations, regarding player lanes, positions, win rate, most played champions, and champions with the most wins (Figure 37). It also has visualizations with data displayed over time for KDA (Figure 38), and the total of minions and jungle monsters kills, Creep Score (CS) (Figure 39). At the top of these visualizations is information about the amount of games considered in the analysis, together with the selected player most played position and champion.

The first visualization techniques available are pie charts, chosen to display statistical information related to roles, lanes, win rate, and KDA. The pie chart was selected for its ability to offer an almost immediate notion of which of the presented attributes has an higher value, by perceiving the size of the respective slice color in the pie. The user can interact with the visualizations by hide or showing attributes, for instance, in the KDA pie the assists can be hidden creating a direct comparison between kills an deaths. The user can also hover the mouse on each pie slice to see the exact value representing that portion.

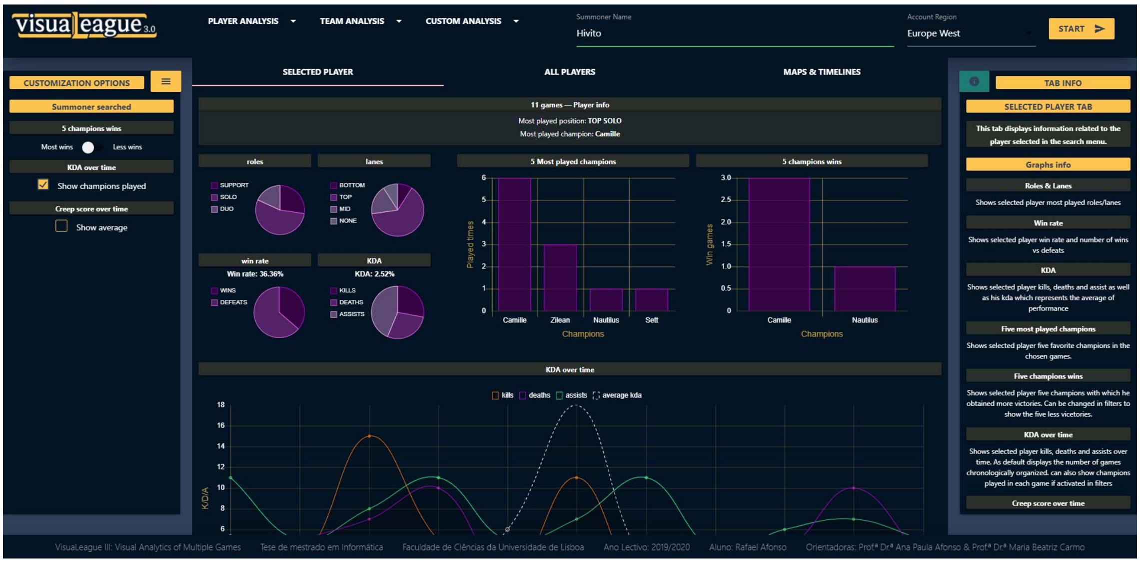


Figure 37: VisuaLeague III player analysis aggregated interface. Selected Player option.

Most played champions, and champions with the most wins are represented in bar charts visualizations since it is important to see exactly how many occurrences each case has, while also perceive the most common and uncommon cases. These visualizations are presented side by side since it is important to relate them in the analysis. For instance, it is possible to analyse the example given by the professional coach (Section 3.4), where he states the importance of knowing if the player's most played champion is what offers the most victories. The user can hover the mouse through each bar to display exact values if needed. It is possible to activate the filter related with five champion wins to display champions with less win ratio.

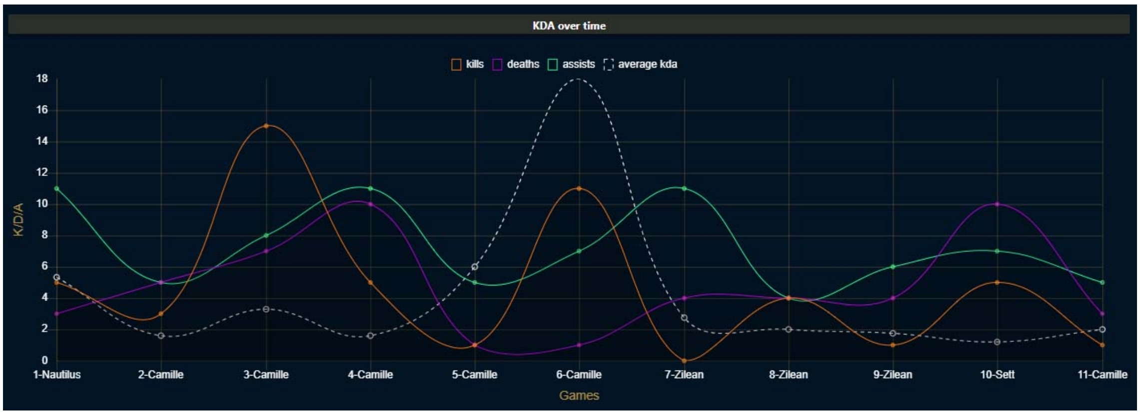


Figure 38: Player analysis aggregated interface — KDA evolution over time.

Figure 38 shows data over time through a line chart. In this case, the horizontal axis represents each game and the user can visualise each KDA attribute, kills, deaths and assists of the player or team, for each match. Champions played on each game can be displayed in the side panel filters to be directly related with the KDA of each game. This visualization gives the analyst a notion of progression, where s/he can see if the player is stable or has an high deviation in scores, while also seeing with which champion he can obtain better performance. In this case a line chart was the most appropriate due to overlapping of elements. By giving lines thin strokes and opacity to shapes, it is easier for the user to be able to read every occurrence while keeping an overall perspective of performance since everything is visible. The user can interact with this visualization, namely the attributes (kills, deaths, assists, average kda) that s/he wants to analyse. The user can hover the mouse in each value (dot) in the lines to see exact values.

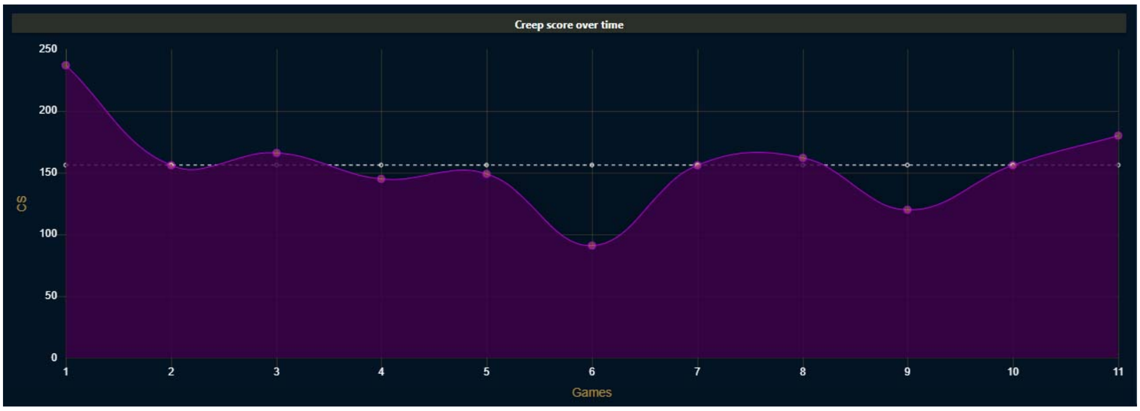


Figure 39: Player analysis aggregated interface — CS evolution overtime.

The graph presented in Figure 39 has the same purpose, but instead of representing the KDA, is the CS that is displayed. The line chart was chosen for its ability to give a good idea of flow enabling the identification of where the amount of CS is constant. KDA and CS visualizations can be directly compared to have complete score information for players' performance as each match is represented in the same position.

All players tab

On the All players tab (Figure 40), is presented information of the players who participated in the selected games. This view is composed by three graphs, namely, most played champions, champions with more wins, and ten most banned champions and respective pick turn for the ban.

In resemblance with the previous tab, most played champions and champions with more wins, are also displayed. The color scheme is different from the Selected player tab (from purple to green) to help the user perceive that the presented data refers to a different matter. With this visualization, the coach can know which are the champions most likely capable of obtaining the victory in the current rank league of the player.

In Figure 41 is a visualization of all bans in loaded games. When preparing a team one of the main concerns of a LoL coach, is to prepare their players' champions against opponents' choices. Champions to pick, or ban, have a major part in this preparation,

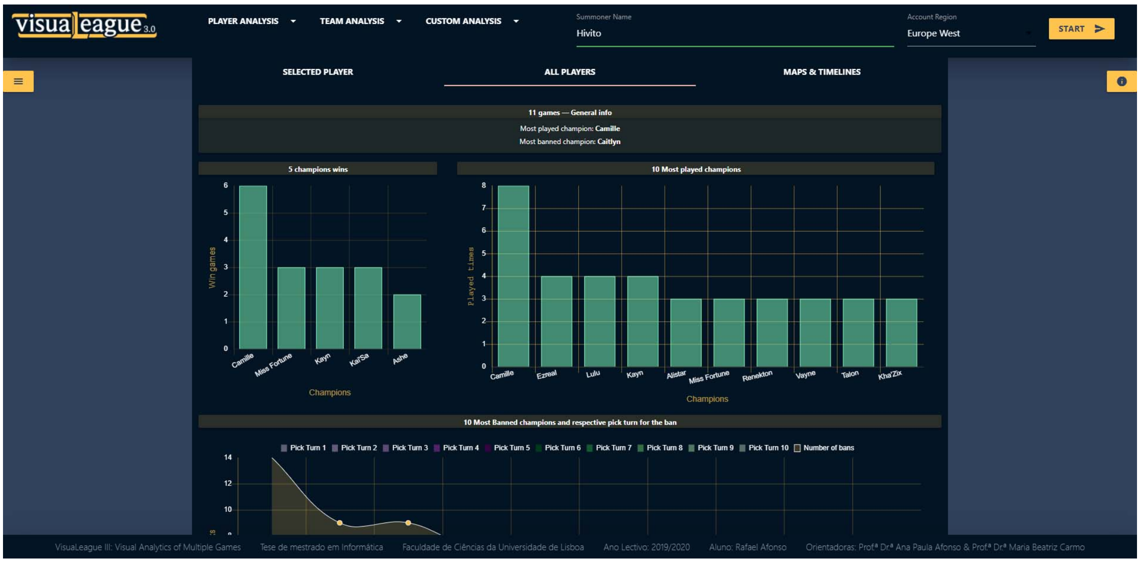


Figure 40: Player analysis aggregated interface — All Players option.

since a given champion is directly countered by another. Knowing how often champions are banned is important, as those are likely to be excluded from the game, making them a less reliable option.

Together with the number of bans is the turn in which each ban occurred. This is useful for analysts to know how early in champion select said champion will be excluded. With this information a team can adapt strategies to select a specific champion, or opt to exclude that option instead.

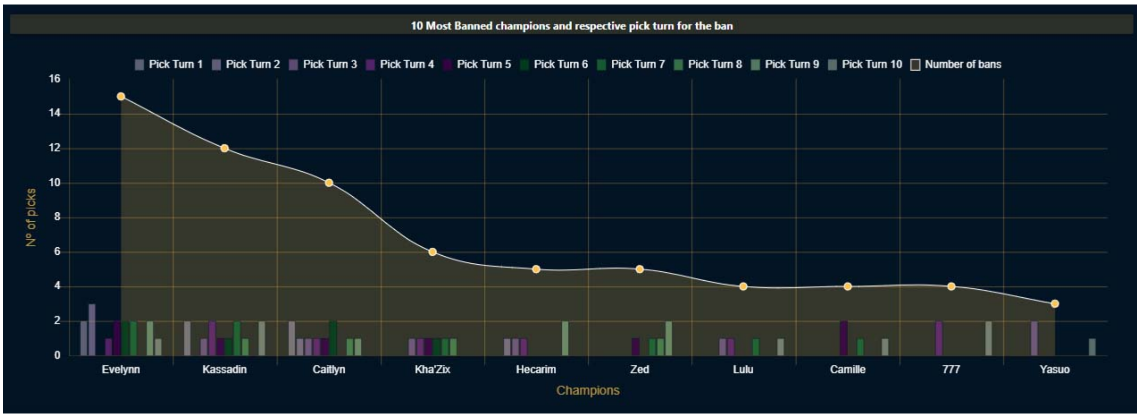


Figure 41: Player analysis aggregated — graph bans and ban turns.

This visualization is a fusion of line and bar charts. The amount of times a champion was banned is represented by the line to take advantage its opacity capabilities, maintaining the readability while overlapping the bars. Since pick turns have a fixed number, they were defined as a set of bars that are displayed in the same vertical alignment as the respective banned champion. The user can interact with the attributes (number of bans or any of the pick turns) hiding non relevant information, for instance, displaying only champions banned in the first pick turn, as they are considered instant bans making those champions unplayable in many matches. The user can hover the mouse on the bars and points in the line to see totals for each case.

Maps and timelines tab

The Maps and timelines tab has a LoL map with representations of several ST information, namely, the position of major events and player positions over matches. The left map in Figure 42 shows every kill event that happened, and the right map represents tower destruction. Two maps are displayed at once and are independent of each other, so the user can easily compare information, for instance, positions every game won versus every game lost. The user can interact with the map by adapting the zoom level and move the current view position. Every event displayed can be interacted by the user, with a mouse click, to display additional information on that specific event. For instance, clicking on a kill event will show information about the players involved.

Figure 42: Player analysis for multiple games (eleven games). Left: Maps with kill events. Right: tower destruction events.

In this interface, the left panel filters are essential to achieve a good analysis. The first filter available (data filters) gives the option to display data for every player, or for the selected player. In team analysis and custom analysis, instead of filtering the data to a selected player, it is filtered to the selected team.

Next, there are four options where it is possible to show and hide wins or defeats for each team side. If wins and red team check boxes are selected, the map will show matches that were won while being on the red team. In player analysis, this filters act on the selected player information. In team analysis, this filters act on the selected team. In custom analysis, this filters are not available due to data limitations.

Next, in the data filters, are the four different types of information available to display. Those can be player movement, player kills, elite monsters (baron, herald and dragon kills) and towers destroyed.

Time data

The time data filter (Figure 42 left panel) gives an option between watching all occurrences in games or choose to use the timeline. Below each map is placed a timeline (Figure 43), which the user can access to control time, capable of going to a specific time and observe the events that occurred in that minute. Figure 43 shows the timeline deactivated at the left and activated at the right. Time frames are available in intervals of five minutes. It was decided to use intervals of five minutes based on the fact that the professional coach, in his analysis of the games, does not analyse all the moments of the match. One exception to this intervals is the herald spawn, this event occurs at a specific time (eight minutes) and is also available in the timeline. The coach also stated that the longer the game goes, the less relevant information becomes, since the players go into phases of the game that are very unpredictable. Normally he focuses the analyse in the first fifteen minutes of the game and sometimes twenty minutes could be relevant since it is where the baron spawns and it is an important objective in the game.

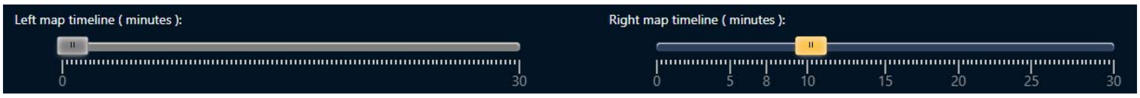


Figure 43: Timeline feature for multiple games analysis. Left: disabled. Right: enabled.

Figure 44 shows the difference of the movement pattern of a top lane player in the first ten minutes of the game and the twenty five minute mark. When the user selects a time, the visualization shows position of players in that exact minute, and events that happened in a range of thirty seconds before or after.

Clustering

Finally in the group points filter (Figure 42 left panel), clustering is used to choose between watching all occurrences individually or in groups created by a clustering algorithm. As the problem of cluttering and overlapping of information are very common in spatiotemporal visualizations of voluminous sets of data, a solution is to use clustering to group points and display the them instead of individuals. It was decided to use the DBSCAN algorithm as it was the better approach to work with density based data, having good outlier perception (Section 2.3.2).

In VisuaLeague III, instead of giving every parameter to DBSCAN to process data, the user has access to the radius slider to interact with the algorithm's Eps, to better fit the analysis needs. A smaller value would be better for analysis displaying a lot of positions,

Figure 44: Player analysis for multiple games. Left: Player positions at ten minute game time. Right: Player positions at twenty five minutes game time.

as it would make positions being considered part of a cluster if they were closer between them. With higher values, more distant positions will be considered part of a cluster, making this a better option when there are less positions being represented in the map. VisuaLeague III will recompute clusters every time the radius slider value is changed, passing a new Eps parameter to DBSCAN and using the center of each cluster to place the position of the group in the map. The size of each circle varies in relation with the amount of positions gathered by that group, resulting in bigger circles for bigger clusters. Figure 45 shows the same data for movement and kills in one hundred games with individual positions in the left map and groups (clustered positions) in the right map. This is an over emphasized example as mentioned before, it is unlikely to load that many games in analysis, but is a good demonstration of the potential to remove clutter in analysis making an almost unreadable map showing points of focus for movement and kills.

4.8 Summary

This chapter presented the VisuaLeague III prototype achieved to analyse League of Legends' matches for three different analysis types offering the option to search games for a single player, teams, professional players, and custom games or scrims.

All the design choices involve the participation of two professional coaches. The prototype architecture was presented together with the methods to extract data, in particular teams and custom matches.

Finally, all the VisuaLeague III types of analysis are described, explaining in detail

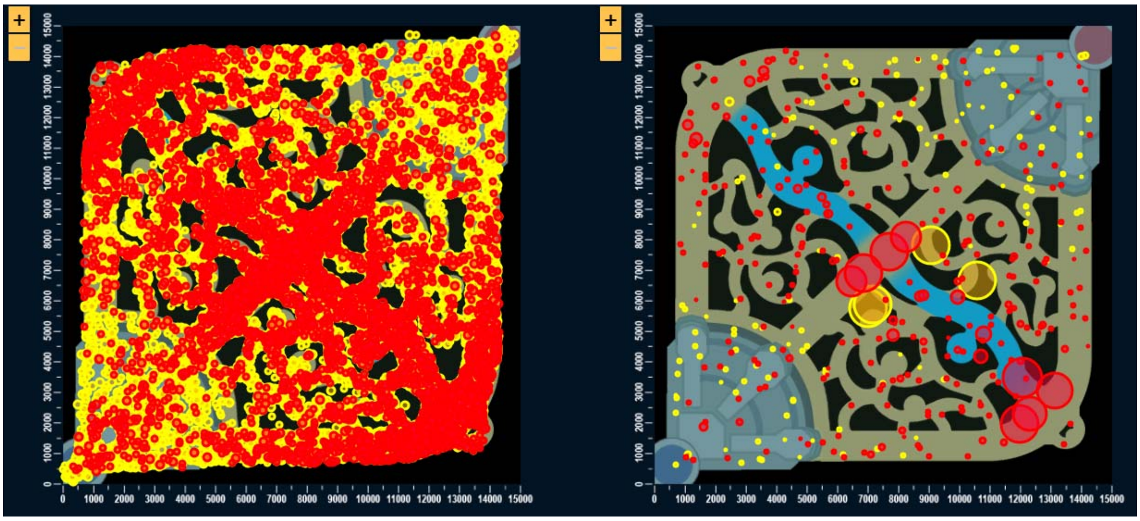


Figure 45: Player analysis for multiple games (one hundred games). Left: Individual positions. Right: clustered positions.

the visualization techniques and how important they are to coach analysis.

Chapter 5

VisuaLeague III — Evaluation

This chapter presents the evaluation of VisuaLeague III' capabilities of providing an analysis tool for League of Legends' competitive matches. Starting by describing the expert participants and methodology in Section 5.1 and respective results in Section 5.2. Finally, a discussion of the results is presented in Section 5.3.

5.1 Participants and methodology

To evaluate VisuaLeague III, interviews (Appendix B) took place via zoom meetings with two League of Legends' professional coaches. One of the coaches has four years of experience with national and international teams. The second coach is currently training the For The Win (FTW) LoL team.

The interviews were made in sessions of approximately three hours and a half. Using the shared screen option and giving control to the application, the coaches were able to use VisuaLeague III. A demonstration of the tool was previously made for each interview as suggested in Ya'acob et al., 2019. After, the analysts were free to explore VisuaLeague III. As the participants are professional coaches, it was asked to perform several analysis for LoL matches for single player, teams and custom matches, using both single and multiple match variants. Additionally, to the participants was asked to perform specific analysis relative to Pandascore's professional team data, visualizations specific to team games, data types and filters used, grouping through clustering, and time filters to better understand analysis needs.

During this process, the observant (author of this thesis) takes notes about the analysis process. Finally, the participants were to fill out a questionnaire with four quality measures, intuitiveness, usefulness, efficiency and innovation for each of the analysis types (Scholtz, 2010). Each measure was classified using a 1-7 scale where one stands for worst and seven stands for best. An optional comments and suggestions field was also provided in the end of each analysis, so the participants could freely share their opinions.

5.2 Evaluation results

The results will be presented starting with single match for single player analysis, Section 5.2.1 for searches focused on a single match from a single player. In Section 5.2.2 are presented results for team analysis while Section 5.2.3 focuses on custom matches analysis, namely, scrims. Finally, Section 5.2.4 shows results for multiple match analysis considering the tree different search methods together, for single, teams and custom games matches as the visualizations are similar.

5.2.1 Player analysis in a single match

In single matches for a single player analysis, the coaches searched for a particular game of interest from the recent match history of the player. In information related to the player specifically the most relevant is related to solo queue rank, while in the matches the focus was on champion, position, game results and KDA. With this information, the coaches chosen a specific match to analyse with the animated map with defeats being, usually, prioritized.

Once in the animated map the tendency was to observe parts of the game where more occurrences were displayed in the timeline.

Figure 46 shows users opinion about this analysis type. We can observe that both have close opinions about usefulness, efficiency and innovation giving those measures high scores. Intuitiveness is the one with more discrepancy as the first user had some difficulties interacting with the animated map timeline and when he paused the time in the timeline, the events disappeared after some time. One of the participants has pointed that it would be useful to include the possibility to search for specific games like in multiple analysis, however, not essential.

5.2.2 Teams analysis in a single match

In terms of team analysis, the coaches started to navigate throughout the available teams to understand what information is useful. All the users considered the Pandascore database not important for the process. However, a more robust database could be considered if the data available was updated more frequently.

Figure 48 shows the opinion related to Pandascore. The first user opted not to give a value as it would be very useful but the data had to be more consistent, while the second user considered the potential of the visualization in case this applies. It was hard to find games for all the selected players from the teams they made with only a few matches after several attempts, however they both found the visualization of the games displayed once those were found very useful, with six and seven as rating.

This analysis type was considered very intuitive, and all the participants assigned a rating seven (see Figure 47). Usefulness and efficiency fall on average values, since it is

Intuitiveness:
How easy was it to understand the analysis process?

Very hard	1	2	3	4	5	6	7	Very easy
User 1			X					
User 2							X	

Usefulness:
How useful was the information extracted from this analysis process?

Useless	1	2	3	4	5	6	7	Very useful
User 1						X		
User 2							X	

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

Takes too long	1	2	3	4	5	6	7	It was fast
User 1					X			
User 2						X		

Innovation:
This analysis method offered something different from the tools you are used to?

Very similar	1	2	3	4	5	6	7	Very innovative
User 1						X		
User 2						X		

Figure 46: Results for single player analysis in a single match.

Intuitiveness:
How easy was it to understand the analysis process?

Very hard	1	2	3	4	5	6	7	Very easy
User 1							X	
User 2							X	

Usefulness:
How useful was the information extracted from this analysis process?

Useless	1	2	3	4	5	6	7	Very useful
User 1					X			
User 2				X				

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

Takes too long	1	2	3	4	5	6	7	It was fast
User 1			X					
User 2						X		

Innovation:
This analysis method offered something different from the tools you are used to?

Very similar	1	2	3	4	5	6	7	Very innovative
User 1						X		
User 2				X				

Figure 47: Results for teams analysis in a single match.

Team games preview:
Tables with champion information and statistics of loaded matches gave a good perception of the matches available?

Need more/other information	1	2	3	4	5	6	7	It is a good preview
User 1							X	
User 2						X		

Pandascore professional teams' data:
Having a database with default professional teams…

Is unnecessary	1	2	3	4	5	6	7	Is helpful
User 1	-	-	-	-	-	-	-	
User 2						X		

Figure 48: Results in a single match using Pandascore's data.

Intuitiveness:
How easy was it to understand the analysis process?

Very hard	1	2	3	4	5	6	7	Very easy
User 1						X		
User 2							X	

Usefulness:
How useful was the information extracted from this analysis process?

Useless	1	2	3	4	5	6	7	Very useful
User 1						X		
User 2							X	

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

Takes too long	1	2	3	4	5	6	7	It was fast
User 1					X			
User 2							X	

Innovation:
This analysis method offered something different from the tools you are used to?

Very similar	1	2	3	4	5	6	7	Very innovative
User 1							X	
User 2						X		

Figure 49: Results for custom analysis match.

hard to find games where professional players play together, in solo queue, and the user must know who played together recently, to find matches.

5.2.3 Custom match analysis

With a custom link given to VisuaLeague's search, the process of adding a name to the search, in order to store data, was counter intuitive. This made the first user give a six to intuitiveness (see Figure 49).

Usefulness and innovation had very high scores with six and seven in both cases, since this is the only tool they know of, capable of displaying this data. The presented visualizations from the match gave a fast and objective analysis, without the need to see the game in the animated map, in some analysis.

Efficiency had a rate of five and seven values. The first user gave a lower score as he thinks it would be more efficient if the graphs shown for gold, damage and minions killed, had been inverted. The radar chart with the KDA was found innovative and very useful to compare two players together.

5.2.4 Multiple match analysis

In this type of analysis, the coaches used different number of games, varying between five, ten, twenty and forty. One of the users explained that he usually considers twenty games for solo queue and less than fifteen for scrims and custom games matches. The game type used was always ranked matches, except in custom games analysis where it must be custom games. The match selection filter where a specific champion is selected to request matches was never used. This could indicate that it is always important to have every champion in perspective when analysing player performance in LoL matches.

Intuitiveness:
How easy was it to understand the analysis process?

Very hard	1	2	3	4	5	6	7	Very easy
User 1							X	
User 2						X		

Usefulness:
How useful was the information extracted from this analysis process?

Useless	1	2	3	4	5	6	7	Very useful
User 1							X	
User 2						X		

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

Takes too long	1	2	3	4	5	6	7	It was fast
User 1							X	
User 2				X				

Innovation:
This analysis method offered something different from the tools you are used to?

Very similar	1	2	3	4	5	6	7	Very innovative
User 1			X					
User 2				X				

Figure 50: Results for multiple matches analysis relative to statistics.

Intuitiveness:
How easy was it to understand the analysis process?

Very hard	1	2	3	4	5	6	7	Very easy
User 1						X		
User 2							X	

Usefulness:
How useful was the information extracted from this analysis process?

Useless	1	2	3	4	5	6	7	Very useful
User 1						X		
User 2							X	

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

Takes too long	1	2	3	4	5	6	7	It was fast
User 1							X	
User 2							X	

Innovation:
This analysis method offered something different from the tools you are used to?

Very similar	1	2	3	4	5	6	7	Very innovative
User 1							X	
User 2							X	

Figure 51: Results for multiple matches analysis relative to map visualizations.

Figure 50 shows the results for multiple games analysis related with statistical information displayed in the visualizations presented in the two first tabs of the interface. Intuitiveness and usefulness have an high rating as it is possible to extract information about a big amount of matches and visualize the most important factors such as scores averages, KDA, champions pick rate and win rate and player performance. However, the first user gave a seven rating for efficiency and the second one gave only four. The reason is related with the fact that the first user is more familiarized with the visualizations, as it is the coach that participated in the design process. In terms of innovation, the presented

Filter by player/team versus everyone:

	User 1	*User 2*
Rather use data showing everyone		
Rather use data for a player/team	***X***	
Both options are important		***X***

Figure 52: Results for focus on analysis comparing searched information and general information.

visualizations were considered with average ratings as they are common when representing this information in other analysis web applications, like OPGG. However, those are usually not applied to multiple games.

Different data types usefulness:

User 1

	1	***2***	***3***	***4***	***5***	***6***	***7***
Movement						***X***	
Kill events						***X***	
Tower kill events					***X***		
Elite monster kill event					***X***		

User 2

	1	***2***	***3***	***4***	***5***	***6***	***7***
Movement							***X***
Kill events							***X***
Tower kill events							***X***
Elite monster kill event							***X***

Different data types usefulness for custom games:

User 1

	1	***2***	***3***	***4***	***5***	***6***	***7***
Movement			***X***				
Kill events						***X***	
Tower kill events						***X***	
Elite monster kill event							***X***

User 2

	1	***2***	***3***	***4***	***5***	***6***	***7***
Movement							***X***
Kill events							***X***
Tower kill events							***X***
Elite monster kill event							***X***

Figure 53: Results for most important event ratings in spatiotemporal data distinguishing solo queue ranked games from custom games and scrims.

The visualizations for ST data were considered much more innovative and efficient, achieving the top score for both users and with a six and seven for usefulness respectively (see Figure 51). The reason for these scores compared to the previous for statistical visualizations was the capability of identify player behaviour in different situations. For instance, related with top lane warding correct positions at the correct time. It was also surprisingly possible to understand player behaviour outside a specific match. One of the users was able to determine that a requested player was dodging games in the blue side, by analysing player events for each map color side, and relating that information with victories and defeats. The term dodging refers to a situation that happens when a player enters the champion selection screen and for some reason his teammates, or team composition, appears to be unplayable for him, making the player quit the game before it starts with a smaller penalty in rating than if he lost the game.

In this interface, both users were overwhelmed by the amount of filters and interactions between them for each of the maps. However, after some minutes they were able to easily use the interface, which they considered to be a fast process compared to the complexity the visualization presents, giving also high scores for intuitiveness with a six and seven, respectively.

Each coach presented different methods of analysis. A very clear indicator is how they considered the filter to display only the selected player, or every player in the game (see Figure 52). One user considers only important to analyse his team, while the other considers every player relevant. Another indicator is how the presented events and movement were rated, relative to the importance in the analysis. One of the users considered every spatial and temporal information to be equally important in custom and solo queue ranked games (see Figure 53). On the other hand, the other coach stated that he analyzes different types of data in different types of games. He considers the movement less important when evaluating custom games, as the players usually know better how to move in the map, and it is uncommon to point flaws on that regard. Also, in custom games, the objectives (towers and elite monster kills) are more relevant, as they are the focus of a team. In solo queue matches, are more common strategies where the player is the focus, making tracking his/her movement more relevant.

Intuitiveness:
How easy was it to understand grouping (clustering)?

Very hard	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***Very easy***
User 1					***X***			
User 2						***X***		

Usefulness:
The ability to group helped understanding the analysis?

Useless	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***Very helpful***
User 1							***X***	
User 2						***X***		

Group radius interaction:

	User 1	*User 2*
Rather not interact with cluster values and let it be an automated process		
Rather have more interactions and options when creating groups		
One simple interaction (as it is now) is enough	***X***	***X***

Figure 54: Results for spatiotemporal data aggregation (clustering).

In terms of aggregation of spatiotemporal data (clustering), Figure 54 shows the rates for the intuitiveness and usefulness measures. Intuitiveness was rated by both coaches

Intuitiveness:
How easy was it to understand timeline's functionality?

Very hard	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Very easy*
User 1							**X**	
User 2							**X**	

Usefulness:
The ability to analyze data in specific timeframes for multiple games is…

Useless	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*Very helpful*
User 1							**X**	
User 2							**X**	

Timeline time stamps:	*User 1*	*User 2*
Rather have timestamps every minute		
Rather have timestamps every time an event or x number of events happen		
The timestamps presented were the most appropriate	**X**	**X**

Figure 55: Evaluation results for time filters (timeline) in spatiotemporal data every five minutes with eight minute being also available due to herald spawn.

with five and six values, respectively. The coaches considered the hardest interaction as it was completely new. However it was not hard for them to understand the relation between the radius value (Eps) and the displayed groups. The ability to group positions was considered very useful with rates of seven and six, respectively. Specially in the analysis with over ten games as the information starts to get very cluttered in lower zoom levels. In terms of options to control the clustering process, they considered the control of the radius sufficient to the analysis. It is important to note that the grouping option was majorly used to get a general notion of the game or player performance. The ungrouped data was also important to give a more detailed view in important moments, like player positions during objectives time.

Figure 55 shows the unanimous opinion from both coaches about the intuitiveness of the timeline functionality and the usefulness of the ability to analyse data in a specific time. The usefulness is rated with seven as analyses focusing on specific time frames are always important, specially in a game like LoL where players have specific timestamps to achieve goals. Intuitiveness was rated with seven values for both cases as the interaction was very easy to understand and utilize. The current time stamps available also proved to be the most appropriate ones for both users. However they stated that it would also be viable to have time options for every minute, for some specific cases, but on the other hand it could imply more work to achieve the same results.

5.3 Discussion

VisuaLeague III proved to be a viable option to analyse LoL matches in the competitive environment of the game. Both coaches subjected to the tests, were able to extract useful information about their professional players in solo queue, search by groups of players and analyse custom matches from their practice games with professional teams.

Following the order of the results presented in Section 5.2, in player analysis in a single match, the coaches were able to analyse the performance of the searched players. The fact that the data from riot API is stored in timestamps of sixty seconds was a limitation for spatiotemporal data visualizations, namely the animated map. It prevented a precise analysis of individual player behaviour, since it is an approximation of player positions. However, it was possible to determine how effective was the player decision making.

A problem that both identified was related to player roles assignment in the animated map. This happens because the data from the match loaded does not consider when players switch lanes at the champion select, or inside the game. A solution for this would be to give user interaction in player roles, so the coaches could change lane position manually throughout the match.

The visualizations presented in team analysis were considered very useful, efficient and intuitive. However, in team search, both coaches tried to find matches for their team players ranked games, but could extract few information. They stated that this was expected, since players usually do not play together, or with the same accounts, outside of scrims and tournaments. They suggested to enable a search for every given summoner account and display matches where any of them participated. Coaches considered the custom analysis a great addition to the analysis tool. Since custom games are important to analyze professional training matches, and other tools do not support this game mode. The visualizations presented were considered the most appropriate for custom analysis, as it displayed the most relevant information to analyze player performance in LoL matches.

The access restrictions imposed on player identities, in custom matches, was expected to be a limitation when performing analysis, but this was not the case. The coaches were able to identify the players without that information being given by VisuaLeague III.

In multiple match analysis, the filters provided for searching matches were appropriate. However, the possibility of searching for games according to date, instead of number of games, would also be valued in some analysis. The visualizations of multiple matches ST data on maps was the most appreciated feature in multiple analysis. It allowed the coaches to take conclusions of player and teams performance, by comparing different game results, team sides and objectives. Both the available maps were used in analysis. A common approach to use this feature was to display different team sides on each map and compare objectives achieved by the player or team. The timeline was considered essential to focus the analyse on specific objectives, and the possibility to group positions through clustering was considered helpful, to remove cluttering and overlapping, specially in analysis with twenty or more matches.

Chapter 6

Conclusion and Future Work

This chapter will conclude this report. Section 6.1 summarizes the information obtained through related work, other League of Legend's web applications, the concept of professional teams in LoL's scope, the main features of VisuaLeague III development process and user opinions about its functionality. In Section 6.2 useful improvements of the current prototype will be discussed as a possible path to follow in future work presented in a global perspective and for each available analysis type.

6.1 Conclusion

A deeper look into the e-sports world helped realise the magnitude of games nowadays and their relevance in the near future. This shows the importance of VisuaLeague as a tool to analyse games data and also reinforce the prominent future the area of virtual games offers.

A search for visualization techniques was conducted to better understand the concept and realize the advantages and disadvantages of several approaches with focus on animated maps, spatiotemporal data and aggregations. User interaction is a very important factor to consider as it allows to filter and manipulate data to provide readability and focus. For aggregations, clustering methods are the most popular ones, being DBSCAN a common option in spatiotemporal analysis which led it to be a part in the implementation of the prototype. Maps and animated maps are a good visualization technique, as they offer the user exact information of where events happened. Those are more relevant together with spatiotemporal data and, if possible, a timeline to manipulate time. However, some statistical information is essential for analysis and will always require support of more simple graphs, like pie charts or bar charts.

A lot of visualisation applications for League of Legends already exist, but none of them offer the same potential as VisuaLeague III, offering the most wanted features of static data analysis, a visual animated representation of individual matches, and options to search for groups of players and custom games information. VisuaLeague III also has options to analyse multiple games for a single player, teams, and scrims, taking advantage of different charts visualizations for statistical data, and maps for ST data.

For single matches, general statistical information is provided to view each game individually. For multiple matches, statistical information is presented as aggregations of matches. Options to filter searches and customize visualizations are available in every analysis type to help the user focus the analyse.

The concept of professional LoL teams and players can be very ambiguous, which led to an implementation of a team database that was then, placed as second plan, while new features for a user to create his own teams were introduced. The protected data of custom games made them also hard to make available, with the help of a professional coach, however, a way to access those datasets and filter them was found in a reliable way. Turning this analysis method one of the most interesting features of the project.

VisuaLeague II was already a complex prototype, requiring several technologies to fully execute, and changes had to be done since the beginning, what made the start of the work more time consuming but also more interesting. New patches for League of Legends are constantly arriving, with new features to be considered in the prototype, new maps, abilities, champions, items, skills and different monsters. In November 2019 the way Riot API delivered data also changed making the process more complex. After understanding and make the previous prototype operational again, the whole layout had to be changed in order to implement new analysis methods, with other ways of gathering data and new visualizations. To gather datasets for teams and multiple matches a lot of pre-processing needs to be done, as Riot API is not prepared to do this searches by itself.

The professional team coaches found the resultant application overall very helpful, with potential to extract good analysis from each of the search methods. It would be impossible to not consider certain limitations like information only being available each sixty seconds, if this was not the case, a lot more could be done with the data, since it would give precise player position over time.

6.2 Future Work

For VisuaLeague III to grow as a tool for scouting, every search could utilise the same data filter menu as the one implemented for multiple matches. This would allow the user to get only game types of interest in every analysis, to improve efficiency in searches. To the already presented filters, time could be added when requesting data for recent matches. This would allow the user to get information for a defined time period in addition to quantity of games, making searches more flexible.

Focusing on single player single match analysis of VisuaLeague III, the overview of searched games could gain relevance if a visualization closer to the one presented in searched team matches was used. This would offer more depth in each game's preview allowing more judgment before the animated map be required. The major limitation of the animated map is directly related with Riot API restrictions. Although some of those cannot be addressed, like data only being kept every sixty seconds, there are some that can be bypassed giving more interactions to the user. For instance when predicting players' lanes in game the interface could allow a user to change player positions, if they change middle game, instead of trying to figure the more adequate one.

Custom game analysis and multiple match analysis are the ones with more potential. The custom games has the advantage of being a tool very useful for coaches and professional players as there is no other option to extract data for those match types. Filters for data related to specific players chosen by an user can be a rewarding path as it would allow to improve every visualization related to multiple matches since the given players could be used to further develop visualizations with specific focus.

The idea of a professional team must be slightly changed in order to bring more utility for professional coaches and players. Team analysis would be more accurate considering teams as groups of players' accounts, where those can be from the same, or different people. This way is possible to provide a search tool that, unlike the current implementation where every selected player must be in the team to find games, the results would show every game where any selected player participated. Posteriorly, additional filters to facilitate and choose groups of matches could exist. This would offer coaches the same analysis potential as the current prototype, while also serve as a search tool for multiple games of different accounts. The resultant search would allow to find more matches, and not require the user to have previous information about teams, making the process of tracking players more efficient.

Multiple games analysis visualizations on statistics would benefit from new additions of time filters. An example of this would be win rate in ranked matches over the last week, as one of the coaches find this information to be very relevant data.

Item related statistics could also be a good addition for multiple match analysis, where the most bought item in those games show for a searched player when he plays a specific champion, for instance.

It would also be possible to show results for statistics of a specific time frame, giving a coach the possibility to chose a time and check if the user is meeting the requirements in that specific time in every game.

For ST data in multiple games, the major limitation is, again, the sixty second intervals limitations from Riot API, as with smaller intervals, it would be possible to make more efficient analysis, and predictions. Improvements in filters is always an option. In multiple matches for single player searches and custom games, filters related to specific lanes could be added, displaying information related to a specific role at a time. Also an option for analysing only data relative to the opposite team in the games could be useful to make team comparisons, as the only two options at the moment is the searched players' team side or everyone.

Despite none of the test cases mentioned it, the timeline could evolve from selecting a specific time frame to enable time intervals selections as well. This way it would be possible to focus the analysis on early, mid, and endgame, serving as a mid term filter for a specific timestamp and every time. The clustering served the purpose of simplifying positions to give better readability, reducing cluttering and overlapping, but it could also be used in other ways. For instance, cluster champions selected with the respective positions over multiple games, to realise where a specific champion usually moves the most. This could prove to be a challenge, however, due to, once again, the sixty second time limitation from the API on movement data.

Bibliography

Afonso, Ana Paula et al. (2019). "VisuaLeague: Player performance analysis using spatial-temporal data". In: *Multimedia Tools and Applications*, pp. 1–22. DOI: 10.1007/s11042-019-07952-z (cit. on p. 6).

Agarwal, Shivam, Günter Wallner, and Fabian Beck (2020). "Bombalytics: Visualization of Competition and Collaboration Strategies of Players in a Bomb Laying Game". In: *Computer Graphics Forum* 39.3, pp. 89–100. DOI: 10.1111/cgf.13965. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13965. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13965 (cit. on pp. 8, 9, 11).

Andrienko, Gennady and Natalia Andrienko (2008). "Spatio-temporal aggregation for visual analysis of movements". In: *Visual Analytics Science and Technology, 2008. VAST'08. IEEE Symposium on*. IEEE, pp. 51–58 (cit. on p. 10).

— (2010). "Interactive cluster analysis of diverse types of spatiotemporal data". In: *Acm Sigkdd Explorations Newsletter* 11.2, pp. 19–28 (cit. on p. 13).

champion.gg (2019). URL: https://champion.gg/ (visited on 11/10/2019) (cit. on p. 23).

ChartJs (2019). URL: https://www.chartjs.org/ (visited on 11/10/2019) (cit. on pp. 24, 34, 36).

Chikish, Yulia, Miquel Carreras, and Jaume Garci (Apr. 1, 2019). "eSports: A New Era for the Sports Industry and a New Impulse for the Research in Sports (and) Economics?" In: pp. 477–508. ISBN: 978-84-17609-23-8 (cit. on pp. 1, 5, 6, 38).

Copeland, Brian et al. (Apr. 25, 2018). "Data Visualization in Games". In: *Major Qualifying Projects (All Years)* (cit. on pp. 7, 14).

CouchDB (2019). URL: https://couchdb.apache.org/ (visited on 11/10/2019) (cit. on pp. 24, 34, 35).

CruzaderKnights2 (2019). URL: https://i.redd.it/lbtz2ilsosi21.jpg (visited on 12/04/2019) (cit. on p. 14).

D3 (2020). URL: https://d3js.org/ (visited on 03/17/2020) (cit. on p. 36).

DataDragon (2019). URL: https://riot-api-libraries.readthedocs.io/en/latest/ddragon.html (visited on 11/13/2019) (cit. on p. 38).

Esport API (2019). URL: https://esport-api.com/ (visited on 11/16/2019) (cit. on p. 39).

express.js (2019). URL: https://expressjs.com/ (visited on 11/10/2019) (cit. on pp. 24, 34, 35).

ExpressNode (2019). *Introdução Express/Node*. MDN Web Docs. URL: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introdu%C3%A7%C3%A3o (visited on 10/31/2019) (cit. on p. 35).

gamepedia (2020). URL: https://www.gamepedia.com/ (visited on 06/10/2020) (cit. on p. 30).

Gamescorekeeper (2019). URL: https://gamescorekeeper.com/ (visited on 11/16/2019) (cit. on p. 39).

gol.gg (2019). URL: https://gol.gg/esports/home/ (visited on 11/10/2019) (cit. on p. 23).

Gonçalves, Tiago, Ana Paula Afonso, and Bruno Martins (2015). "Cartographic visualization of human trajectory data: Overview and analysis". In: *Journal of Location Based Services* 9.2, pp. 138–166 (cit. on p. 9).

Gonçalves, Tiago et al. (July 2018). "Analysing Player Performance with Animated Maps". In: *2018 22nd International Conference Information Visualisation (IV)*. 2018 22nd International Conference Information Visualisation (IV). ISSN: 1550-6037, pp. 103–109. DOI: 10.1109/iV.2018.00028 (cit. on pp. 8, 18).

Guo, Diansheng et al. (Nov. 2006). "A Visual Inquiry System for Space-Time and Multivariate Patterns (VIS-STAMP)". In: *IEEE Transactions on Visualization and Computer Graphics - TVCG* 12 (cit. on p. 9).

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis (2001). "A Cluster Separation Measure". In: *Journal of Intelligent Information Systems* 17.2, pp. 107–145. ISSN: 09259902. DOI: 10.1023/A:1012801612483 (cit. on p. 12).

Hallmann, Kirstin and Thomas Giel (Feb. 1, 2018). "eSports – Competitive sports or recreational activity?" In: *Sport Management Review* 21.1, pp. 14–20. ISSN: 1441-3523. DOI: 10.1016/j.smr.2017.07.011 (cit. on p. 5).

Hamari, Juho and Max Sjöblom (2017). *What Is eSports and Why Do People Watch It?* SSRN Scholarly Paper ID 2686182. Rochester, NY: Social Science Research Network (cit. on p. 1).

He, Jing et al. (2019). "Variable-Based Spatiotemporal Trajectory Data Visualization Illustrated". In: *IEEE Access* 7, pp. 143646–143672. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2942844 (cit. on p. 2).

Hirschberg, Daniel S. (Oct. 1977). "Algorithms for the Longest Common Subsequence Problem". In: *J. ACM* 24.4, pp. 664–675. ISSN: 0004-5411. DOI: 10.1145/322033.322044 (cit. on p. 11).

History of Cartography (2017). *History of Cartography: 2nd Edition (Paperback) - Routledge*. Routledge.com. URL: https://www.routledge.com/History-of-Cartography-2nd-Edition/Bagrow/p/book/9781412811545 (cit. on p. 7).

Holden, John T., Anastasios Kaburakis, and Ryan Rodenberg (2017). "The Future Is Now: Esports Policy Considerations and Potential Litigation". In: *Journal of Legal Aspects of Sport* 27, p. 46 (cit. on p. 1).

Hu, Tianxiao et al. (Feb. 2020). "AntVis: A web-based visual analytics tool for exploring ant movement data". In: *Visual Informatics* 4. DOI: 10.1016/j.visinf.2020.02.001 (cit. on p. 6).

jDBSCAN (2020). URL: https://github.com/upphiminn/jDBSCAN (visited on 04/20/2020) (cit. on pp. 34, 36).

Jiang, Xingyu et al. (2020). "Robust Feature Matching Using Spatial Clustering With Heavy Outliers". In: *IEEE Transactions on Image Processing* 29, pp. 736–746. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2019.2934572 (cit. on p. 13).

Journal, Ijcsmc (Jan. 1, 2019). "A Comparative Study of the Growth of Electronic Sports in the World and the Important Global E-Sports Achievements". In: *IJCSMC* (cit. on p. 5).

Kanagala, H. K. and V. V. Jaya Rama Krishnaiah (2016). "A comparative study of K-Means, DBSCAN and OPTICS". In: *2016 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–6. DOI: 10.1109/ICCCI.2016.7479923 (cit. on p. 13).

Kuan, Yen-Ting, Yu-Shuen Wang, and Jung-Hong Chuang (Oct. 2017). "Visualizing Real-Time Strategy Games: The Example of StarCraft II". In: *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2017 IEEE Conference on Visual Analytics Science and Technology (VAST). ISSN: null, pp. 71–80. DOI: 10.1109/VAST.2017.8585594 (cit. on pp. 7, 8).

LCS (June 14, 2011). *Longest Common Subsequence — DP-4*. GeeksforGeeks. URL: https://www.geeksforgeeks.org/longest-common-subsequence-dp-4/ (visited on 11/15/2019) (cit. on p. 11).

Leaflet (2019). URL: https://leafletjs.com/ (visited on 11/10/2019) (cit. on pp. 24, 34, 36).

Lol Tier List (2019). *Lol Tier List: Top Challenger Meta Picks by Role (NEW Patch 9.22)*. Mobalytics. URL: `https://app.mobalytics.gg/lol-tier-list` (visited on 11/11/2019) (cit. on pp. 20, 21).

lolcounter (2019). URL: `https://lolcounter.com/` (visited on 11/10/2019) (cit. on p. 23).

lolpros (2020). *LOLPros.GG*. URL: `https://lolpros.gg/` (visited on 06/10/2020) (cit. on pp. 20, 29).

Markus Persson (2008). *Infinite Mario Bros.* Andy G's Blog. URL: `https://gieseanw.wordpress.com/tag/infinite-mario-bros/` (visited on 11/27/2019) (cit. on p. 10).

Materialize (2019). URL: `https://materializecss.com/` (visited on 11/11/2019) (cit. on pp. 24, 34, 36).

Mejorado, D. A. M. and J. E. R. Marquez (2019). "Visual analytics of team performance in online gaming". In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3025–3030 (cit. on pp. 8, 10).

MOBAFire (2019). URL: `https://www.mobafire.com/` (visited on 11/11/2019) (cit. on p. 23).

Mobalytics (2019). *Mobalytics - Analytics for gamers*. URL: `https://app.mobalytics.gg/champions` (visited on 11/11/2019) (cit. on pp. 22, 23, 29).

Moucho, Tiago Alexandre Orrico (2018). "VisuaLeague II - Animated maps for performance analysis in games". In: (cit. on pp. 1, 2, 8, 15, 25, 27).

Nanni, Mirco and Dino Pedreschi (Nov. 2006). "Time-Focused Clustering of Trajectories of Moving Objects". In: *J. Intell. Inf. Syst.* 27, pp. 267–289. DOI: `10.1007/s10844-006-9953-7` (cit. on p. 13).

Neo4j (2019). URL: `https://neo4j.com/` (visited on 11/10/2019) (cit. on pp. 24, 25, 34, 35).

Newn, Joshua et al. (2017). "Evaluating Real-Time Gaze Representations to Infer Intentions in Competitive Turn-Based Strategy Games". In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. CHI PLAY '17. event-place: Amsterdam, The Netherlands. New York, NY, USA: ACM, pp. 541–552. ISBN: 978-1-4503-4898-0. DOI: `10.1145/3116595.3116624` (cit. on pp. 14, 15).

node.js (2019). URL: `https://nodejs.org/en/` (visited on 11/10/2019) (cit. on pp. 24, 34).

noUiSlider (2019). URL: `https://refreshless.com/nouislider/` (visited on 11/10/2019) (cit. on pp. 24, 34, 36).

Olsen, Anders Hval (Sept. 28, 2015). “The Evolution of eSports: An Analysis of its origin and a look at its prospective future growth as enhanced by Information Technology Management tools”. In: *arXiv:1509.08795 [cs]*. arXiv: 1509.08795 (cit. on p. 1).

OPGG (2019). *LoL Stats, Record Replay, Database, Guide*. URL: https://euw.op.gg/ (visited on 11/10/2019) (cit. on p. 29).

Probuilds (2019). *League of Legends Champion Pro Builds and Guides*. URL: https://www.probuilds.net/ (visited on 11/10/2019) (cit. on pp. 21, 22, 29).

Riot Games (2019a). *League of Legends*. URL: https://euw.leagueoflegends.com/en/ (visited on 11/12/2019) (cit. on p. 17).

— (2019b). *Riot Developer Portal - Summoner by name*. URL: https://developer.riotgames.com/apis#summoner-v4/GET_getBySummonerName (visited on 11/12/2019) (cit. on p. 44).

Rogers, Kevin B. and G. C. White (2007). “Analysis of Movement and Habitat Use from Telemetry Data”. In: (cit. on p. 6).

Schertler, Ruben, Simone Kriglstein, and Günter Wallner (Oct. 17, 2019). “User Guided Movement Analysis in Games using Semantic Trajectories”. In: pp. 613–623. ISBN: 978-1-4503-6688-5. DOI: 10.1145/3311350.3347156 (cit. on pp. 1, 7, 9, 11).

Scholtz, Jean (2010). “Developing qualitative metrics for visual analytic environments”. In: *Proceedings of the 3rd BELIV’10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization*, pp. 1–7 (cit. on p. 63).

scikit-learn (2019). *Machine learning in python*. URL: https://scikit-learn.org/stable/index.html (visited on 11/14/2019) (cit. on p. 12).

Seif, George (Sept. 14, 2019). *The 5 Clustering Algorithms Data Scientists Need to Know*. Medium. URL: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68 (visited on 11/14/2019) (cit. on p. 12).

Shi, Zhicheng and Lilian S. C. Pun-Cheng (Mar. 2019). “Spatiotemporal Data Clustering: A Survey of Methods”. In: *ISPRS International Journal of Geo-Information* 8.3, p. 112. DOI: 10.3390/ijgi8030112 (cit. on pp. 7, 12, 13).

Skobtsov, Vadim et al. (2017). “Intelligent Telemetry Data Analysis of Small Satellites”. In: *Cybernetics and Mathematics Applications in Intelligent Systems*. Ed. by Radek Silhavy et al. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, pp. 351–361. ISBN: 978-3-319-57264-2. DOI: 10.1007/978-3-319-57264-2_36 (cit. on p. 6).

Stefani, Raymond (Mar. 15, 2017). “E-sports, mind sports and the Olympics: What is a sport, anyway?” In: *Statslife-Significance Magazine* (cit. on p. 5).

Thompson, Joseph J. et al. (Sept. 18, 2013). "Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning". In: *PLOS ONE* 8.9, e75129. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0075129 (cit. on pp. 5, 6).

Tobler, Waldo R (1987). "Experiments in migration mapping by computer". In: *The American Cartographer* 14.2, pp. 155–163 (cit. on p. 9).

topEsports (2019). *Top Games Awarding Prize Money - Esports Game Rankings :: Esports Earnings*. URL: https://www.esportsearnings.com/games (visited on 11/12/2019) (cit. on p. 1).

Tork, Hadi Fanaee (2012). "Spatio-temporal clustering methods classification". In: *Doctoral Symposium on Informatics Engineering*. Vol. 1. 1. Faculdade de Engenharia da Universidade do Porto Porto, Portugal, pp. 199–209 (cit. on p. 13).

u.gg (2019). URL: https://u.gg/ (visited on 11/10/2019) (cit. on p. 23).

Vieira, Pedro Miguel Almeida (2017). "Visualization of spatio-temporal information for personal performance analysis in games". In: (cit. on pp. 2, 8).

Wallner, Guenter (July 1, 2018). "Automatic generation of battle maps from replay data". In: *Information Visualization* 17.3, pp. 239–256. ISSN: 1473-8716. DOI: 10.1177/1473871617713338 (cit. on p. 11).

Wallner, Günter (June 11, 2019). "Enhancing Battle Maps through Flow Graphs". In: *arXiv:1906.04435 [cs]*. arXiv: 1906.04435 (cit. on pp. 7, 9, 12).

Wallner, Günter, Nour Halabi, and Pejman Mirza-Babaei (2019). "Aggregated Visualization of Playtesting Data". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. event-place: Glasgow, Scotland Uk. New York, NY, USA: ACM, 363:1–363:12. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300593 (cit. on pp. 1, 9, 10, 13).

Ya'acob, Suraya et al. (2019). "VISUAL ANALYTICS EVALUATION PROCESS: PRACTICE GUIDELINES FOR COMPLEX DOMAIN". In: *Malaysian Journal of Computer Science*, pp. 118–134 (cit. on p. 63).

Appendix A

Coaches interview

1. **Que tipo de dados costumas usar na análise? Explica-me um bocado o teu processo.**

   Focam-se mais em preparar a equipa consoante picks e escolhas.

2. **Quais os dados que extrais das ferramentas de análise (OPGG, etc)?**

   Costumam servir mais para jogadores não conhecidos: ver os champions mais jogados, win ratio dos champions mais jogados, KDAs. Winning ratios dos jogadores, posições dos jogadores.

   As ferramentas permitem pesquisar pela equipa e saber todas as contas do jogador.

   Permite ver se o jogador está a jogar com os Champions indicados.

3. **Costumas analisar jogadores individualmente, a equipa ou ambos? Há critérios de análise para uma equipa/jogador individual que sejam independentes dos adversários.**

   Ver se compram items certos na altura certa. Altura em que vão à base. Se vão à base todos juntos ou não (se não foram todos juntos atrasa porque não podem fazer jogadas de equipa durante mais tempo).

4. **Costumas analisar um jogo inteiro, ou partes especificas dum jogo? Determinas isso por tempo in-game ou por eventos?**

   Depende dos jogos. Procura normalmente dar track ao jungler, ele é o mainshot caller. Rotações mid game. Não vê o jogo todo, mas procura eventos específicos. É importante ver quase tudo até aos 8 minutos porque é quando aparece o herald.

5. **Quais são os fatores determinantes para que uma equipa seja considerada ter uma boa performance? Como costumas determinar isso?**

   Se vê um jungler a dar invade sem prio (ter perioridade das lanes) claramente está a jogar mal o jogo. O jungler só deve invadir se tiver as lanes puxadas ou souber exatamente onde está o jungler. Se esse conceito não está bem definido, a equipa, normalmente não está bem preparada. Também acontece com os suportes. Com a altura que escolhem para colocar wards. Também são apanhados a dar invade. É visível o nível da equipa também quando a equipa dá engage em alturas erradas para a composição que tem. Por ex.: uma equipa má para early engage dar invade logo. Uma boa equipa escolhe as fights certas, normalmente por um objectivo (baron, dragon).

6. **Haverá alguma vantagem em analisar vários jogos apenas de um jogador? Ou é irrelevante se não houverem dados do resto da equipa e/ou adversários?**

   Normalmente conhecer o jogador e treinar a equipa são coisas distintas. Os treinos são feitos em scrims, jogos privados.

7. **No contexto de treinar uma equipa, cada jogador pratica uma posição fixa. Costumas comparar jogadores de posições opostas simultaneamente (por ex.: jungler vs jungler)? Se sim, isso acontece levando em conta toda a equipa ou só os dois jogadores?**

   A posição individual é relevante. Normalmente sabe-se à partida se o nosso jungler é melhor ou pior do que o inimigo, o objetivo é arranjar uma estratégia que ajude o mais fraco. Uma estratégia que esconda as fraquezas. Por ex.: um jogador mais fraco joga com um tank, para aguentar a pressão.

8. **Em média quantos jogos analisas para tirar conclusões? Algum critério específico para escolher os jogos a analisar?**

   Em competitivo costumamos analisar os últimos 4-5 jogos da equipa. No entanto percursos da equipa num grande número jogos seria útil. Não adianta ver jogos muitos atrás porque se ganharem 30 jogos sem perder é porque a estratégia está a resultar então basta ver os últimos 5 e analisar a estratégia, por exemplo.

   É relevante comparar padrões de uma equipa que ganhou muitos jogos com um especifico em que perdeu, por exemplo. Para ver o que mudou nas decisões tomadas.

   Em competitivo, de uma equipa não faz muito sentido 100 jogos mas de uma liga talvez, mas mesmo assim... Não por uma equipa. 15-20 jogos é o aceitável. Esta split teve 14 jogos e 3 best of five em playoffs. No máximo uma equipa jogaria 29 jogos.

   Para um jogador individual analisar o seu desempenho já é diferente. Já faria sentido 100-200 jogos e torna-se mais importante representar o movimento deste, já que é ele que está a jogar.

   Seria relevante uma ferramenta que analisasse 100 jogos e ele poder escolher um tempo específico para ver onde houve mais movimento para saber tendências. Tanto ou mais importante que o tempo pode ser eventos de destruição de torres ou matar dragões.

9. **Que informação achas relevante aceder em simultâneo se pudesses ver esses jogos todos juntos? Gold? Percursos? Kills? Objetivos?**

   Movimento de jungler é o mais importante de traquear. Heatmaps de Wards são muito importantes. O que acontece depois de uma torre ser destruída ou algum evento relevante acontecer.

   A que tempos é que as equipas costumam fazer o drake e o herald. (os primeiros são mais relevantes).

10. **Há alguma coisa que costumes sentir falta nas ferramentas que utilizas para análise?**

    Ferramenta para as scirms...

    Heatmaps de wards por tempo. Conseguir saber o q as pessoas estão a wardar, a q tempo.

    Nos drafts de uma equipa. Que Champions é que foram banidos para determinado lado do campo por equipa. Por exemplo eles banem sempre Aphelios. Perceber as picks de uma equipa. Ver o que costumam pickar e banir, e fazer um levantamento do que eles querem.

    Prio é muito importante. Prio pode ser por ex., a capacidade de um champion dar clear.

    Perceber se o champion é para escalar ou se é para pressionar ao inicio.

Appendix B

Experts evaluation tests

B.1 Test case 1

Single match analysis

Search and analyze single player games

Process description:

Execution (how the user did it):

Searched for a professional player giving attention focus to rank, lane in each game, victory, KDA and game type. The chosen game to further analyze was a defeat in top lane (the main lane of the player) to realize what went wrong. Started the visualization with the animated map where analyzed gold, builds and events giving more time where there was more information displaying in the timeline.

Analysis (what the user did):

Analysis of a single game for a single player. Understanding the most interesting game to further analyzing from recent match history. Proceed to a more in-depth analysis with the animated map, watch player performance by stats, scores, and positions. Compare players of same lanes one against the other in terms of gold collected and stats.

User opinion about single player matches analysis process:

Intuitiveness:

How easy was it to understand the analysis process?

	1	*2*	*3*	*4*	*5*	*6*	*7*	
Very hard			***X***					*Very easy*

Usefulness:
How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless						***X***		*Very useful*

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long					***X***			*It was fast*

Innovation:
This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar						***X***		*Very innovative*

Additional thoughts on single match analysis (optional):

Filters to the games searched could be implemented such as the ones in multiple game matches.

When pausing the animated map displayed events should be kept. Had some troubles watching match total time and current time, felt like I had to tur my attention away from the match to realize current time. The 60 seconds timestamp are a limiting factor sometimes. Could have the ability to swap champion positions when interacting with each one in stats tab.

Single match analysis

Search and analyze team games (group of players)

Process description:

Execution (how the user did it):
Started by scrolling in already built teams understanding what could be done. Then proceeded to build a new one, found some difficulty understanding the interaction to search for the team and only one player. Searched for players from his team but was a hard task as they don't usually play together, and when they do, they often use "smurfs". Proceeded to analyze games from other players who he knew played together. The user didn't feel the need to use the animated map in this case as the game preview gives more detailed information.

Analysis (what the user did):
Analysis of games from a group of players. Understanding how practical is to search for players as a team. Analyze player performance in a game they played together and understand if, on average, the team tends to win or lose.

Team games preview:

Tables with champion information and statistics of loaded matches gave a good perception of the matches available?

	1	2	3	4	5	6	7	
Need more/other information							**X**	*It is a good preview*

Pandascore professional teams' data:

Having a database with default professional teams…

	1	2	3	4	5	6	7	
Is unnecessary	-	-	-	-	-	-	-	*Is helpful*

User opinion about team games analysis process:

Intuitiveness:

How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard							**X**	*Very easy*

Usefulness:

How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless					**X**			*Very useful*

Efficiency:

How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long			**X**					*It was fast*

Innovation:

This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar						**X**		*Very innovative*

Additional thoughts on team analysis (optional):
Having a frequently updated database with professional teams would be an amazing feature. Pandascore's database is not enough as of this point in time.

It would be better if the analysis served as a finder for people that play together instead of searching for people I knows played together recently, for instance: put every account of the players I knows and detect every game two or more of them participated.

The visualization for each game could be used in the single player as well.

Single match analysis

Analyze custom games

Process description:

Execution (how the user did it):
The user searched for a custom match link (link is more usual than game id), the name for the search was counter intuitive at the first try. The analysis of the preview was rather extensive, being the focus player builds, bans, gold and KDA. User compared two champions of the same role together in addition to the rest of the team. The animated map visualization was started after this analysis to try to understand exactly what happened. User jumped to near the end of the game to watch player movement in other to complement the analysis. However there was some absent information due to custom matches restrictions they weren't even noticed until mentioned.

The focus of the analysis was on the lane of the player from who the link was extracted, elite monster kills, and tower destruction are key factors in scrims. Constantly keeping eyes on every player items at certain time frames.

Analysis (what the user did):
Understanding player performance in a scrim game using a private link to realize key factors that contributed to a victory/defeat.

User opinion about custom games analysis process:

Intuitiveness:
How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard						***X***		*Very easy*

Usefulness:
How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless						***X***		*Very useful*

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long					***X***			*It was fast*

Innovation:
This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar							***X***	*Very innovative*

Additional thoughts on custom games analysis (optional):
Very good to have analysis for custom games as there are no current tools to get this information.
In the graphs the second team would be better if was inverted for more intuitive analysis.
Every information is very useful, the animated map visualization has the throwback of having only timestamps each minute which can make the analysis unreliable because we're watching players buy a certain item at that time, but as the gold they have is only updated every minute there's no way to know if the player is making the best choice or not.

Multiple matches analysis

Process description:

Execution (how the user did it):
The user analyzed various amount of games for various players. The number of matches requested vary among five, ten or twenty games being five the first impulse to search and explaining that twenty games is the usual when analyzing multiple games. The game type used was always ranked matches (except in custom games analysis where it must be custom game). After analyzing five games however, the user felt the necessity of getting more, as the player was having a lot of defeats in recent matches, requesting then the twenty. The user was able to get player's performance information with the presented graphs giving special attention to KDA, bans, win rate and champion with most wins. Realized that the player in question was having a mediocre performance in recent games overall and even that he left one of the matches (probably because of some technical difficulty).

After knowing this information, proceeded to analysis using the maps and timelines tab. Here the kill events were the focus for a while using most of the time data filtered to the searched user or team. The user could understand differences in player behavior when he was on different teams and if the game was won or defeated, the first reaction being "This player is hinting a lot" he then realized that there was a lot more wins in one team than another which was important to find what can be done better when playing that side. The timeline was occasionally used normally in the eight, ten and fifteen timestamps (somewhere in the mid game) where the fight for the objectives start to happen and a-lot of decisive choices occur.

The user felt the need of cluster positions when not filtering to the searched player data but when analyzing a specific time frame the clustering option was always disabled.

The user searched majorly for places where the player killed/died the most, elite monsters killed when winning and losing games for each team side. This events near the timestamp of the herald (8 minutes) are very important.

Elite monsters and tower kills are very important for custom games, while movement and kills are more relevant for solo queue matches.

The user got a general idea of the game using clustering but to deeply understand the game usually he rather uses individual positions with a high zoom level.

Just by watching positions the user was able to decipher the current meta of the game, realizing that it was currently mid/bottom lanes which made sense with his idea what made him think that it could be used to analyze Korean players to “search” for the next meta. Also immediately understanded that one of the players being searched haven’t got a single game while playing the blue side of the map making him think that was likely that the player searched was “dodging” the queue when he got a game in the blue side.

Analysis (what the user did):
User searched for multiple games to get a notion of player performance in recent matches in ranked games for objective completion, movement/kills in certain time frames, Victory and defeats for each team side of the map.

Charts, statistics:

User opinion about charts and statistics analysis process:

Intuitiveness:
How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard							**X**	*Very easy*

Usefulness:
How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless							**X**	*Very useful*

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long							**X**	*It was fast*

Innovation:
This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar			**X**					*Very innovative*

Additional thoughts on multiple games' charts and statistics analysis (optional):
A graph containing gold could also be added, but the CS graph compensated for it in a way.

Maps & timelines:

Different data types usefulness:

	1	2	3	4	5	6	7
Movement						**X**	
Kill events						**X**	
Tower kill events					**X**		
Elite monster kill event					**X**		

Different data types usefulness for custom games:

	1	2	3	4	5	6	7
Movement			**X**				
Kill events						**X**	
Tower kill events						**X**	
Elite monster kill event							**X**

Filter by player/team versus everyone:

Rather use data showing everyone	
Rather use data for a player/team	**X**
Both options are important	

Grouping player positions (clustering):

Intuitiveness:
How easy was it to understand grouping (clustering)?

	1	2	3	4	5	6	7	
Very hard					**X**			*Very easy*

Usefulness:
The ability to group helped understanding the analysis?

	1	2	3	4	5	6	7	
Useless							***X***	*Very helpful*

Group radius interaction:

Rather not interact with cluster values and let it be an automated process	
Rather have more interactions and options when creating groups	
One simple interaction (as it is now) is enough	***X***

Additional thoughts on grouping, clustering (optional):
Clusters could separate kills from deaths when searching for players, such like when not clustered. Grouping positions is surely useful when the maps gets filled with positions and kill events.

Timeline data:

Visualization aggregating time in intervals of five minutes

Intuitiveness:
How easy was it to understand timeline's functionality?

	1	2	3	4	5	6	7	
Very hard							***X***	*Very easy*

Usefulness:
The ability to analyze data in specific timeframes for multiple games is…

	1	2	3	4	5	6	7	
Useless							***X***	*Very helpful*

Timeline time stamps:

Rather have timestamps every minute	
Rather have timestamps every time an event or x number of events happen	
The timestamps presented were the most appropriate	***X***

Additional thoughts on time aggregations (optional):
Timestamps for each minute would also be fine, it would give more customization which is always good, but I still think the current five minutes are enough and more simple.

User opinion about Maps & timelines analysis process:

Intuitiveness:

How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard						***X***		*Very easy*

Usefulness:

How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless						***X***		*Very useful*

Efficiency:

How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long							***X***	*It was fast*

Innovation:

This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar							***X***	*Very innovative*

Additional thoughts on multiple matches analysis (optional):

The innovation would be even greater if counting only custom games, since this is the only tool available to get data from this type of games.

Being able to do this for every other country (namely Korea) would be very interesting to find new patters for the new meta.

B.2 Test case 2

Single match analysis

Search and analyze single player games

Process description:

Execution (how the user did it):

Searched for a professional player giving attention focus to lane in each game, KDA and game type. Analyzed more than one game, victory and defeat to see differences. Started the visualization with the animated map where analyzed builds and movement and events. Skipped the first 2-3 minutes of the game.

Analysis (what the user did):

Analysis of a single game for a single player. Understanding player performance in recent match history. Proceed to a more in-depth analysis with the animated map, watch player performance by positions over time and builds.

User opinion about single player matches analysis process:

Intuitiveness:

How easy was it to understand the analysis process?

	1	*2*	*3*	*4*	*5*	*6*	*7*	
Very hard							***X***	*Very easy*

Usefulness:

How useful was the information extracted from this analysis process?

	1	*2*	*3*	*4*	*5*	*6*	*7*	
Useless							***X***	*Very useful*

Efficiency:

How fast could you get to what you need for the analysis (extract useful information)?

	1	*2*	*3*	*4*	*5*	*6*	*7*	
Takes too long						***X***		*It was fast*

Innovation:

This analysis method offered something different from the tools you are used to?

	1	*2*	*3*	*4*	*5*	*6*	*7*	
Very similar						***X***		*Very innovative*

Additional thoughts on single match analysis (optional):

Player lanes/positions seem changed when they switch in champion select.

Single match analysis

Search and analyze team games (group of players)

Execution (how the user did it):
Tried to search games from one player in two distinct accounts, making a “team” with his Summoner name and region. Visualeague did not find matches. As for groups of different players matches were found but the usefulness of the information was limited as players often don’t play together in solo queue. Proceeded to see presented information in a group of players for normal games where team composition and statistics were the main focus.

Analysis (what the user did):
Analysis of matches for a group of different accounts from the same player. Analysis of games for a group of solo queue players.

Team games preview:

Tables with champion information and statistics of loaded matches gave a good perception of the matches available?

	1	2	3	4	5	6	7	
Need more/other information						*X*		*It’s a good preview*

Pandascore professional teams’ data:

Having a database with default professional teams…

	1	2	3	4	5	6	7	
Is unnecessary						*X*		*Is helpful*

User opinion about team games analysis process:

Intuitiveness:
How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard							*X*	*Very easy*

Usefulness:
How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless				*X*				*Very useful*

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long						***X***		*It was fast*

Innovation:
This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar				***X***				*Very innovative*

Additional thoughts on team analysis (optional):

I think that the team-match analysis has limited functionality and could probably get merged with something else in order to make a more viable scouting resource.

Single match analysis

Analyze custom games

Process description:

Execution (how the user did it):
With a given custom link tried to extract information. Every presented graph seemed to have similar relevance with the exception of damage meters that was less used. Didn't feel the need to go for the animated map in this case.

Analysis (what the user did):
Analysis of a custom match from a link to understand what happened in that match and if the team was victorious for some particular reason.

User opinion about custom games analysis process:

Intuitiveness:
How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard							***X***	*Very easy*

Usefulness:
How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless							***X***	*Very useful*

Efficiency:
How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long							***X***	*It was fast*

Innovation:
This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar						***X***		*Very innovative*

Additional thoughts on custom games analysis (optional):

____
____
____
____
____

Multiple matches analysis

Charts, statistics:

Process description:

Execution (how the user did it):
The user analyzed various amount of games from ten to thirty, he noted that having the option to search for games according to time instead of number of games was also good to have. The user was able to get player's general performance special attention to win rate, most played champions and champion with most wins. Felt a necessity for more graphs like, last seven days win rate by champions.

In the maps tab the user gives similar relevance to every data presented. Tried to understand where the team was focusing more and having more fights considering each different game result and team side. The timeline was used in objectives timestamp, in this case heralds. Clustering was necessary special when analyzing data for both teams in thirty games as the amount of presented data was very large.

Analysis (what the user did):
User searched for multiple games understand where the teams spent more time and why.

User opinion about charts and statistics analysis process:

Intuitiveness:

How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard						**X**		*Very easy*

Usefulness:

How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless						**X**		*Very useful*

Efficiency:

How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long				**X**				*It was fast*

Innovation:

This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar				**X**				*Very innovative*

Additional thoughts on multiple games' charts and statistics analysis (optional):

Multiple team analysis could use some reworking on as a tool, it's pretty intuitive but It'd need some changes in order to be usable as a tool for scouting.

Multiple matches analysis

Maps & timelines:

Different data types usefulness:

	1	2	3	4	5	6	7
Movement							**X**
Kill events							**X**
Tower kill events							**X**
Elite monster kill event							**X**

Different data types usefulness for custom games:

	1	2	3	4	5	6	7
Movement							X
Kill events							X
Tower kill events							X
Elite monster kill event							X

Filter by player/team versus everyone:

Rather use data showing everyone	
Rather use data for a player/team	
Both options are important	X

Grouping player positions (clustering):

Intuitiveness:

How easy was it to understand grouping (clustering)?

	1	2	3	4	5	6	7	
Very hard						X		*Very easy*

Usefulness:

The ability to group helped understanding the analysis?

	1	2	3	4	5	6	7	
Useless						X		*Very helpful*

Group radius interaction:

Rather not interact with cluster values and let it be an automated process	
Rather have more interactions and options when creating groups	
One simple interaction (as it is now) is enough	X

Additional thoughts on grouping, clustering (optional):

____

____

____

____

____

Timeline data:

Visualization aggregating time in intervals of five minutes

Intuitiveness:

How easy was it to understand timeline's functionality?

	1	2	3	4	5	6	7	
Very hard							**X**	*Very easy*

Usefulness:

The ability to analyze data in specific timeframes for multiple games is…

	1	2	3	4	5	6	7	
Useless							**X**	*Very helpful*

Timeline time stamps:

Rather have timestamps every minute	
Rather have timestamps every time an event or x number of events happen	
The timestamps presented were the most appropriate	**X**

Additional thoughts on time aggregations (optional):

____
____
____
____
____

User opinion about Maps & timelines analysis process:

Intuitiveness:

How easy was it to understand the analysis process?

	1	2	3	4	5	6	7	
Very hard							**X**	*Very easy*

Usefulness:

How useful was the information extracted from this analysis process?

	1	2	3	4	5	6	7	
Useless							**X**	*Very useful*

Efficiency:

How fast could you get to what you need for the analysis (extract useful information)?

	1	2	3	4	5	6	7	
Takes too long							***X***	*It was fast*

Innovation:

This analysis method offered something different from the tools you are used to?

	1	2	3	4	5	6	7	
Very similar							***X***	*Very innovative*

Additional thoughts on multiple matches analysis (optional):

____
____
____
____
____